Evolutionäres Prototyping multimodaler, farbgraphischer Benutzerschnittstellen in der Medizin - Entwurf, Implementierung und Ansätze zur Bewertung

Von der Fakultät für Elektrotechnik der
Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines
DOKTOR - INGENIEURS
genehmigte Dissertation.

Vorgelegt von
Diplom-Ingenieur

Manfred Heinrich Langen

aus Bonn

Referent: Univ.-Prof. Dr. rer. nat. G. Rau
Korreferent: Univ.-Prof. Dr. rer. nat. D. Haupt

Tag der mündlichen Prüfung: 8. Februar 1991

Inhaltsverzeichnis

Die vorliegende Arbeit entstand während meiner Tätigkeit als Doktorand in der Forschungsgruppe "Ergonomie in der Medizin" am Helmholtz-Institut für Biomedizinische Technik an der Rheinisch-Westfälischen Technischen Hochschule Aachen.

An dieser Stelle möchte ich mich bei allen meinen Kollegen und den Mitarbeitern der Klinik für Anästhesiologie des Klinikums Aachen, die mich bei dieser Arbeit unterstützt haben, herzlich bedanken.

Herrn Univ.-Prof. Dr. rer. nat. G. Rau und Herrn Univ.-Prof. Dr. med. S. Effert, den Direktoren des Instituts, danke ich für die Möglichkeit der Durchführung dieser Arbeit und die stetige Förderung und Betreuung.

Herrn Univ.-Prof. Dr. rer. nat. D. Haupt, dem Leiter des Rechenzentrums und Inhaber des Lehrstuhls für Betriebssysteme der RWTH Aachen, danke ich für das Interesse an dieser Arbeit und die Übernahme des Korreferats.

Herrn Univ.-Prof. Dr. med. G. Kalff, dem Direktor der Klinik für Anästhesiologie am Klinikum Aachen, danke ich für die freundliche Unterstützung bei den Studien zur Gestaltung des Anästhesie-Entscheidungsunterstützungs Systems.

Meinen Kollegen Herrn Dipl.-Inform. B. Thull, Herrn Dipl.-Ing. H.-J. Popp und Herrn Dipl.-Ing. Th. Schecke, dem Leiter der Arbeitsgruppe, danke ich für zahlreiche inhaltliche Diskussionen dieser Arbeit und das fruchtbare Arbeitsklima.

Herrn Dipl.-Des. W. Korr danke ich für die anregenden Diskussionen über Farbgestaltung und die Anfertigung vieler in dieser Arbeit verwendeten Zeichnungen.

Den Diplomanden und studentischen Hilfskräften der Forschungsgruppe danke ich für ihre wertvolle Mitarbeit bei der programmtechnischen Realisierung der in dieser Arbeit beschriebenen Software-Entwicklungsumgebung.

Der Studienstiftung "Cusanuswerk" danke ich für die finanzielle Förderung dieser Arbeit und die persönliche Förderung durch zahlreiche interdisziplinäre Fortbildungsveranstaltungen.

Liste der verwendeten Abkürzungen

AIS	Anästhesie Informationssystem. Computersystem zur Patientenüberwachung und Dokumentation während der Anästhesie (Klocke et al. 1987).
AES-2	Anästhesie Entscheidungsunterstützungs System. Wissensbasiertes System für die Beurteilung des Patientenzustands und die Generierung von Therapievorschlägen (Schecke et al. 1988).
CIE	Commision Internationale de l'Eclairage. International anerkannte Organisation für Standardisierungen in Beleuchtungsfragen.
IFIP	International Federation for Information Processing
OOD	Objektorientiertes Design
OOP	Objektorientierte Programmierung
TCP/IP	Transmission Control Protokoll / Internet Protokoll. Weit verbreitetes Protokoll der Datenübertragung in Rechnernetzen.
UIMS	User Interface Management System
X-11	X-Windows Graphiksystem, Version 11

In dieser Arbeit neu eingeführte Abkürzungen

CEDI	Colour Editor. Software-Werkzeug zur Unterstützung der Farbgestaltung von Benutzerschnittstellen.
DIC	Descriptive Interface Code. Diese Form von Quellkode basiert auf dem Entwurf des Interface Baukasten IBIS.
IBIS	Interface Baukasten für interaktive Schnittstellen. Die Bausteine dieses Baukastens können mit wenig Programmieraufwand zu einer interaktiven Schnittstelle zusammengesetzt werden.

0. Einleitung

Die wachsende Zahl von Diagnose- und Therapiemöglichkeiten in der Medizin wirft das Problem zunehmender Beanspruchung von Ärzten und medizinischem Personal bei der Patientenüberwachung auf. Einen Lösungsansatz kann hier der Einsatz von interaktiven Informationssystemen bieten. Die steigende Leistungsfähigkeit von moderner Hardwaretechnologie bei gleichzeitig sinkenden Preisen erlaubt in zunehmendem Maße den Einsatz von Computersystemen in diesem Bereich. Die Akzeptanz und Effizienz eines solchen Systems ist sehr stark von der Gestaltung der Mensch-Maschine-Schnittstelle abhängig. Gerade im medizinischen Bereich sind die Anforderungen an die Gestaltung der Mensch-Maschine Schnittstelle besonders hoch, weil hier das medizinische Personal als Benutzer dieser Systeme keine technisch orientierte Tätigkeit und daher auch keine technische Ausbildung besitzt.

Dem Bedarf an interaktiven Systemen steht ein Mangel an Richtlinien für die Gestaltung von Mensch-Maschine Systemen gegenüber. Da die Ansprüche des Benutzers immer auf die Erfüllung einer bestimmten Aufgabe hin untersucht werden müssen, wird heute überwiegend eine heuristische Vorgehensweise angewendet. Erfahrungsgemäß steigt der Schwierigkeitsgrad der Systementwicklung umso stärker, je einfacher eine Benutzerschnittstelle erlernbar und fehlerfrei bedienbar sein soll. Verschiedene Studien schätzen, daß künftig für die Mensch-Maschine-Schnittstelle zwischen 30 und 80 % der erforderlichen Kosten einer Systementwicklung aufgewendet werden (Rowe und Shoens 1983). Die Unterstützung des Entwicklers durch geeignete Methoden und Werkzeuge ist jedoch noch unzureichend. Insbesondere ist bisher die Bewertung einer Benutzerschnittstelle während des Gestaltungsprozesses wenig berücksichtigt worden; quantitativ bestimmbare Größen für die Bewertung fehlen entweder ganz oder geben kaum Aufschluß über Ursachen und Verbesserungsmöglichkeiten einer unbefriedigenden Systemleistung. Einflußfaktoren wie Individualität des Benutzers (oder der Benutzergruppe) oder aufgabenbezogene Rahmenbedingungen sind analytisch nicht vollständig erfaßbar und können derzeit effektiv nur in einem heuristischen Vorgehen berücksichtigt werden.

Die beste Methode, Benutzerschnittstellen hoher Akzeptanz und Güte zu erhalten, besteht daher zur Zeit darin, eine iterative Entwicklung in enger Zusammenarbeit mit den späteren Benutzern der Schnittstelle durchzuführen (Budde und Kuhlenkamp 1984); diese Vorgehensweise wird als "partizipatives Prototyping" bezeichnet. In der vorliegenden Arbeit wird diese Form von "Prototyping" auf die schrittweise Entwicklung der softwaretechnisch lösbaren Gestaltungsaufgaben einer Benutzerschnittstelle bezogen. Die Vorgehensweise allein ist jedoch keine Garantie für benutzer-

freundliche Systeme. Die Voraussetzung für ein effektives und erfolgreiches "Prototyping" ist ein umfassendes Konzept für eine leistungsfähige Entwicklungsumgebung, d.h. der technologische Fortschritt in diesem Bereich besteht "in der Entwicklung von Softwareentwicklungsumgebungen, die sämtliche Phasen der Software-Erstellung durch den Einsatz aufeinander abgestimmter, rechnergestützter Werkzeuge unterstützen" (Hübner et al. 1987).

Ziel dieser Arbeit ist es, die Entwicklung von leistungsfähigen Mensch-Maschine-Schnittstellen dadurch zu unterstützen, daß eine Entwicklungsumgebung die iterative Vorgehensweise im Prototyping unter Einschluß ergonomischer Gestaltungsziele berücksichtigt, d.h. verschiedene Aspekte der ergonomischen Gestaltung sollen bereits beim Systementwurf durch geeignete Werkzeuge behandelt werden können. Diese Werkzeuge sollen dem Enwickler einer Benutzerschnittstelle - nachfolgend Systementwickler genannt - helfen, die Gestaltung von Gesamtsystemen unter Berücksichtigung der Anforderungen der späteren Benutzergruppe durchzuführen. Als Benutzer wird nachfolgend ein "typischer Benutzer" angenommen, dessen Verhalten bezüglich einer Arbeitsaufgabe kaum vom Durchschnitt der Benutzergruppe abweicht (z.B. ein Anästhesist während einer Operation oder eine Krankenschwester bei der Intensivpflege).

Die Problematik der Entwicklung einer Benutzerschnittstelle wird in Kapitel 1 aus verschiedenen Blickwinkeln betrachtet: das medizinische Umfeld, die systemergonomischen Aspekte und die Möglichkeiten des Software-Engineering sollen im Ansatz berücksichtigt werden. Als geeignete Vorgehensweise wird in Kapitel 2 in das "evolutionäre Prototyping" eingeführt und das Konzept der in dieser Arbeit beschriebenen Entwicklungsumgebung festgelegt. Die Basis der Entwicklungsumgebung, ein objektorientiertes Baukastensystem für multimodal-interaktive Schnittstellen, wird in Kapitel 3 beschrieben. Dieser Baukasten erfüllt die Funktionalität eines Benutzerschnittstellen Management Systems ("user interface management system" = UIMS). Die darauf aufbauende Werkzeugumgebung umfaßt Editoren für den interaktiven Entwurf der Informationsdarstellung (Kapitel 4) sowie Editoren für die Interaktion mittels Spracherkennung und die Ausgabe von akustischen Meldungen (Kapitel 5). In dieser Entwicklungsumgebung sind erstmals auch Ansätze zur Bewertung von Benutzerschnittstellen integriert.

In Kapitel 6 wird die Gestaltung der Benutzeroberfläche eines Anästhesie-Entscheidungs-Unterstützungssystems beschrieben, die mit Hilfe der vorgestellten Vorgehensweise und Werkzeuge entworfen und implementiert wurde.

1. Der Entwicklungsprozeß von Benutzerschnittstellen

Die Entwicklung von Benutzerschnittstellen erfordert die Berücksichtigung von Methoden aus unterschiedlichen Fachdisziplinen. Die Ergonomie ist ein multidisziplinäres Wissenschaftsgebiet, das sich mit der Interaktion zwischen dem Menschen und der von ihm geschaffenen und gestalteten technischen Umgebung befaßt (Bernotat 1977). Durch den zunehmenden Einsatz von Computern gewinnen Methoden des Software Engineering auch im Bereich der Ergonomie an Bedeutung. Neben diesen Methoden müssen für eine Anwendung im medizinischen Bereich zusätzlich fachspezifische Anforderungen berücksichtigt werden, die aus dem medizinischen Umfeld heraus entstehen.

1.1 Besondere Merkmale des medizinischen Umfelds

Die Zunahme der Komplexität von Instrumenten in der Medizin verläuft in ähnlicher Weise wie in Bereichen der Prozeßüberwachung (Rau und Trispel 1982; Bernotat und Rau 1980). Auch einzelne Arbeitsaufgaben sind durchaus vergleichbar: so beschreibt Gravenstein (1982) die Gemeinsamkeiten der Aufgabe eines Anästhesisten bei der operativen Patientenüberwachung und der Aufgabe eines Piloten bei der Flugzeugführung. In beiden Systemen - Patient bzw. Flugzeug - muß eine Vielzahl von Parametern überwacht und gesteuert werden, wobei ein Fehler lebensgefährdend sein kann. Die Systemsicherheit steht daher im Vordergrund dieser Systeme. Dennoch sind die ergonomischen Anforderungen an die Mensch-Maschine Schnittstelle nicht gleich. Im medizinischen Umfeld müssen einige besondere Randbedingungen beachtet werden (Rau 1979):

- die nichttechnisch orientierte Tätigkeit des medizinischen Personals,
- die Versorgung des Patienten steht im Mittelpunkt.

Ein neu konzipiertes computergestütztes System soll einen direkten Nutzen für den Patienten (z.B. eine bessere Therapie) und/oder eine Arbeitserleichterung für das Personal einbringen. Damit die Neueinführung eines Systems vom Personal mitgetragen wird, muß zumindest ein Mehraufwand an Arbeit vermieden werden; Systeme, die einen zusätzlichen Zeitbedarf durch die Bedienung hervorrufen, werden in der Regel auf Dauer nicht akzeptiert, eventuell sogar boykottiert. Zusammenfassend kann festgestellt werden, daß die Anforderungen an die schnelle und einfache Handhabung sowie leichte Erlernbarkeit in der Medizin oft höher sind als in anderen Bereichen der Prozeßüberwachung.

Ein Lösungsansatz für die Entwicklung neuer Systeme in der Medizin ist die Integration von Informationsdarstellung und Interaktion auf einem berührempfindlichen Farbdisplay (Rau 1979). Bei der Gestaltung soll der Arzt als Benutzer soweit einbezogen werden, daß er nicht nur "sich das mal ansieht" sondern Benutzerbeteiligung intensiv in allen Phasen eines Projekts verwirklicht wird. In der Zusammenarbeit von Ingenieuren und Medizinern müssen zwei Diskrepanzen überbrückt werden:
- Ingenieure und Mediziner verwenden unterschiedliche Fachsprachen,
- die Formulierung und Diskussion von Problemen erfolgt auf unterschiedlichen Abstraktionsebenen.

Infolgedessen sind Spezifikationsverfahren, wie sie in den Ingenieurwissenschaften als Pflichtenheft oder formale Sprache angewandt werden, nicht unmittelbar einsetzbar.

1.2 Aspekte der Systemergonomie

Die Systemergonomie befaßt sich mit der Analyse, Synthese und Bewertung eines Mensch-Maschine-Systems mit dem Ziel, die Leistung dieses Gesamtsystems zu optimieren (Döring 1986). Die Aufgabenteilung zwischen Mensch und Maschine soll die Fähigkeiten des Menschen und die Möglichkeiten einer Maschine optimal kombinieren. Die Eignung des Menschen bzw. eines Computers wird folgendermaßen beschrieben (Wickens 1984):

Der Mensch
- hat die Fähigkeit, völlig neuartige Lösungswege zu gehen,
- definiert Ziele, Teilprobleme und deren Beziehung,
- verfügt über Erfahrung und umfangreiches Allgemeinwissen und kann daher auch Analogieschlüsse ziehen,
- beherrscht eine schnelle Mustererkennung,
- ist als Teilsystem hochadaptiv.

Ein Computer
- ist eine leistungsfähige externe Gedächtnishilfe durch Speicherung großer Mengen numerischer Daten,
- leistet Berechnungen nach einem vorgegeben Schema,
- hat ein konsistentes Verhalten.

Die Kopplung beider Systeme erfolgt über eine Mensch-Maschine-Schnittstelle (bzw. Benutzerschnittstelle), so daß die Qualität dieser Schnittstelle einen wesentlichen Einfluß auf die Leistung des Gesamtsystems hat. Eine große kognitive Anstrengung des Menschen bei der Bedienung einer Maschine ist ein Hinweis auf eine "Inkompatibilität"

zwischen Mensch und Maschine (Hutchin et al. 1986), d.h. der Mensch ist gezwungen, sich an die Maschine anzupassen. Die Aufgabe der Ergonomie besteht in der "Anpassung der informatorischen Anforderungen an die Grenzen und Gesetzmäßigkeiten der Informationsverarbeitung beim Menschen durch geeignete Informationsorganisation und -darstellung" (Kraiss 1985). Daher steht die Berücksichtigung der kognitiven, perzeptiven und motorischen Fertigkeiten des Benutzers im Vordergrund einer systemergonomischen Gestaltung.

Die Begriff der *Qualität* einer Mensch-Maschine-Schnittstelle -oft unspezifisch als "Benutzerfreundlichkeit" bezeichnet- wird in DIN 66234 (1980) genauer spezifiziert:
- Aufgabenangemessenheit
  Die Schnittstelle soll die Arbeitsaufgabe unterstützen, ohne eine zusätzlich Belastung durch die Bedienung zu erzeugen.
- Selbsterklärungsfähigkeit
  Die Bedienung soll unmittelbar verständlich sein bzw. auf Verlangen ("HELP-Funktion") erläutert werden können.
- Steuerbarkeit
  Die Dialogmöglichkeiten und deren zeitlicher Rahmen sollen vom Benutzer bestimmt werden.
- Verläßlichkeit
  Das Verhalten soll der Erwartung des Benutzers entsprechen.
- Fehlertoleranz
  Trotz Fehleingaben soll das Arbeitsziel erreichbar bleiben.
- Erlernbarkeit
  Keine dicken Handbücher oder lange Schulungen sollen notwendig sein.
- Übersichtlichkeit
  Die Information auf dem Bildschirm soll übersichtlich gestaltet sein.
- Flexibilität
  Eine Anpassung an individuelle Ziele und Wünsche soll leicht möglich sein.

Diese Anforderungen können bei bestehenden Systemen als Grundlage für eine Bewertung dienen; dies ist z.B. im System EVADIS durch einen Fragebogen realisiert worden (Murcher et. al. 1987). Dem Entwickler von Mensch-Maschine-Systemen bietet dies jedoch keine konkrete Hilfe. Aus diesem Grund wurde von Seiten der Ergonomie die Anwendung von Software Engineering Methoden schon früh gefordert (Draper & Norman 1984), während die Gestaltung der Mensch-Maschine-Schnittstelle in der Literatur über Software Engineering wenig Beachtung findet (Grudin 1987).

1.3 Aspekte des Software Engineering

Der Begriff des Software Engineering wurde erstmals bei der Garmisch Partenkirchener Nato-Konferenz im Jahre 1968 verwendet. Heute wird der Begriff von der IEEE wie folgt definiert: "... ein systematischer Ansatz zur Entwicklung, Anwendung, Wartung und Verbesserung von Software". Software Engineering umfaßt Verfahren, Werkzeuge und Methoden, die im Idealfall nicht nur untereinander, sondern auch auf eine vorgegebene Problemstellung abgestimmt sind (Pressman 1987).

Die Disziplin des Software Engineering sollte die damals erkannte (und bis heute bestehende) "Software-Krise" bewältigen. Die Software-Krise wird durch folgende Symptome beschrieben (Pressman 1987):
• dramatische Kostenexplosion bei der Entwicklung von Software,
• gesetzte Zeitpläne werden fast nie eingehalten,
• Qualität umfangreicher Software-Systeme ist mangelhaft,
• mangelnder Überblick bei großen Projekten,
• Software Entwicklung kann mit der Hardware-Entwicklung nicht Schritt halten,
• Wartungskosten verschlingen einen Großteil des geplanten finanziellen Etats.

Als Reaktion auf diese Symptome werden heute Anforderungen an die Qualität von Software gestellt, die nach verschiedenen Merkmalen gegliedert werden können (NTG 1986):
- Portabilität:
  Eignung des Programms zur Übertragung auf unterschiedliche Hardware und/oder Softwareumgebungen.
- Zuverlässigkeit:
  Die Eigenschaft eines Programms, eine Funktion unter bestimmten Bedingungen stets für eine vorgegebene Zeitdauer auszuführen.
- Effizienz:
  Der Bedarf an Hard- und Software Ressourcen zur Erfüllung einer vorgegebenen Anforderung (Laufzeit, Speicherbedarf).
- Korrektheit/Fehlerfreiheit:
  Anteil der im Programmsystem enthaltenen Fehler durch falschen Entwurf oder falsche Kodierung.
- Robustheit:
  Eignung des Programms, bei fehlerhafter Konfiguration (Hard- und Software) oder fehlerhaften Eingaben geeignet zu reagieren.
- Wartbarkeit:
  Eignung des Programms für das Erkennen und Beseitigen von Fehlerursachen

(Transparenz, Modularität).
- Anpassbarkeit:
  Aufwand einer Programmänderung aufgrund der Änderung von Anwenderanforderungen (Flexibilität).
- Wiederverwendbarkeit:
  Möglichkeit, Teile des Programmsystems (weitgehend) ungeändert in anderen Softwaresystemen einzusetzen.

Diese Merkmale sind nicht unabhängig voneinander, sondern besitzen gegenseitige Abhängigkeiten; so ist z. B. die Wiederverwendbarkeit mit der Portabilität positiv, mit der Effizienz hingegen negativ korreliert.

Der erste wichtige Beitrag des Software Engineering erfolgte in Form der phasenorientierten Projektmodelle ("Life-Cycle" bzw. Phasen-Modell), deren erste Variante das statische Phasenmodell (Boehm 1976) ist (Bild 1.1).

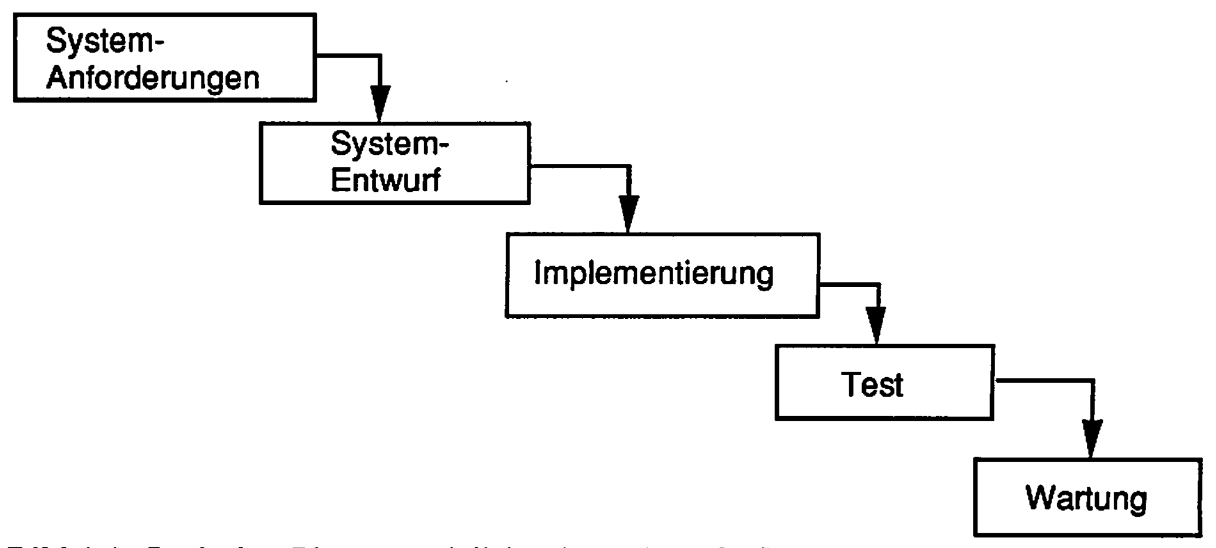


Bild 1.1: Statisches Phasenmodell (nach Boehm 1976).

Dieses Phasen-Modell basiert auf den Annahmen (Floyd 1984), daß:
1. eine vollständige Spezifikation schon zu Beginn ermittelt werden kann und dann festgeschrieben bleibt,
2. ein System einmal hergestellt und dann nur noch gewartet, aber nicht weiterentwickelt wird,
3. umfangreiche Dokumente zur Verständigung über Software zwischen Entwicklern und Benutzern ausreichen.

Diese Annahmen werden heute wegen ihrer Starrheit in Frage gestellt oder für falsch gehalten (Floyd 1984).

zu 1.: Während des Entwicklungsprozesses können Veränderungen in den Anforderungen der Arbeitsaufgabe oder im Leistungsumfang verfügbarer Hardware und den Software-technischen Grundlagen auftreten. Daher muß eine Modifikation oder Anpassung der Software leicht möglich sein.
zu 2.: Bereits Anfang der 80er Jahre wurde erkannt, daß die Annahme in sich abgeschlossener Phasen für die Softwareentwicklung nicht angemessen ist.

> "No matter where you are in the system life cycle, the system will change and the desire to change it will persist throughout the life cycle" (Bersoff 1980).

zu 3.: Bei Systemen mit interaktiver Mensch-Maschine-Schnittstelle sind Dokumente für die Verständigung (Spezifikation) unzureichend, weil:

- die Anforderungen zu Beginn nicht exakt erfaßbar sind,
- keine formale Spezifikationssprache angewendet werden kann, weil diese vom nichttechnischen Benutzer in der Regel nicht verstanden wird (Harker 1987),
- Expertenwissen der Benutzergruppe (Expertenbeteiligung) miteinbezogen werden muß, das nicht am grünen Tisch oder aus Büchern erschlossen werden kann,
- eine inhomogene, nichttechnisch geschulte Benutzergruppe vorliegt.

Infolgedessen können komplexe Softwaresysteme nicht mit dem statischen Phasenmodell bearbeitet werden. Die Forschung entwickelt daher zunehmend prozeßorientierte Ansätze, die die Software nicht mehr statisch als einmal zu spezifizierendes Produkt betrachten, sondern als iterativ entwickelte Folge von Systemversionen (Floyd 1981).

1.4 Prototyping Verfahren

Die iterative Vorgehensweise berücksichtigt im Projektablauf Rücksprünge zwischen den einzelnen Phasen des statischen Phasenmodells. Es werden verschiedene Arten iterativer Vorgehensweisen unter dem Begriff "Prototyping" zusammengefaßt, so daß die in Bild 1.2 eingezeichneten Rücksprünge exemplarische Bedeutung haben. "Prototyping" wird als die "4. Generation" der Software-Entwicklungstechniken betrachtet (Case 1985), nachdem in der 3. Generation Pascal, Fortran, und Prolog zur strukturierten Programmierung innerhalb des statischen Phasenmodells eingesetzt worden waren. Diese beiden Generationen müssen jedoch nicht unabhängig nebeneinander stehen. Vielmehr wird künftig eine sinnvolle Kombination von Prototyping, strukturierter Programmierung und traditionellem Phasen-Modell angestrebt (Pressman 1987, Case 1985).

Spitta (1989) gibt in seiner Mängelliste in bezug auf die Verfahren der 3. Generation u.a. folgende Punkte an:

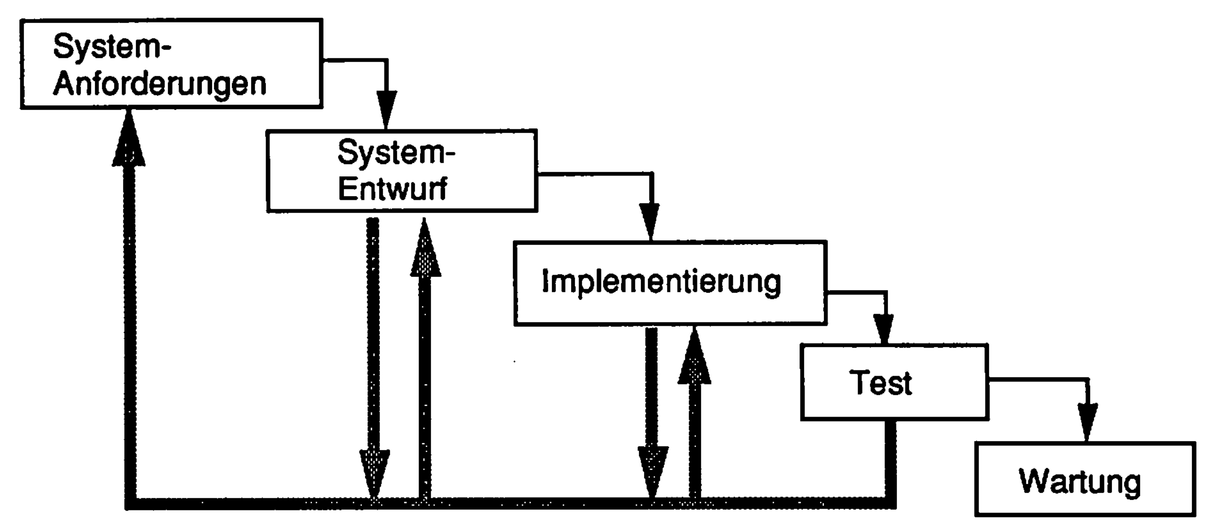


Bild 1.2: Phasenmodell mit Prototyping.

- keine Durchgängigkeit der Methoden über alle Phasen,
- fehlende, mangelhafte oder zu teure Werkzeugunterstützung,
- keine Unterstützung eines Dialogentwurfs,
- kein modularer Entwurf im Sinne des Geheimnisprinzips (Parnas 1972),
- keine Bildung von Teilsystemen.

Diese Einschränkungen können auch für Prototyping Verfahren nur überwunden werden, wenn eine Software-Entwicklungsumgebung auf dieses Verfahren hin abgestimmt wird.

1.4.1 Klassifikation von Prototyping Verfahren

Floyd (1984) führt folgende Systematik ein, die den Sammelbegriff "Prototyping" näher klassifiziert:

Exploratives Prototyping:
Hier wird Prototyping als Möglichkeit benutzt, die Anforderungen an ein Software-Projekt zu klären (Fairley 1985). Dabei werden oft nur Teilaspekte betrachtet. Für die Softwareprototypen ist eine niedrige Zuverlässigkeit und schwache Performance ausreichend, da sie nach Klärung technischer Fragen, z.B. der Gestalt des Benutzerdisplays oder der Dialogstruktur, "weggeworfen" werden. Das Prototyping erfolgt also nur in der Phase der Spezifikation. Dadurch kann der Tatsache Rechnung getragen werden, daß Änderungen in frühen Phasen eines Projekts nur 1 bis 10 % der Geldsumme kosten, die während der Wartungsphase anfallen würden (Pressman 1987).

Experimentelles Prototyping:
Experimentelles Prototyping wird eingesetzt, um einen Lösungsansatz zu prüfen bevor eine umfassende und aufwendige Implementation beginnt. Im Unterschied zum explorativen Prototyping geht diese Vorgehensweise über die Spezifikationsphase hinaus, z.B. zur Analyse des Laufzeitverhaltens. Die experimentellen Implementierungen sind wieder als "Wegwerf-Produkt" anzusehen.

Evolutionäres Prototyping:
Evolutionär bedeutet hier, daß eine fortwährende Anpassung an die sich ändernden Anforderungen über alle Phasen hinweg berücksichtigt wird. Dabei entstehen Versionen, deren Funktionalität iterativ ansteigt. Ein System erreicht normalerweise nie die Phase, wo es als "fertig" angesehen wird; es treten lediglich Pausen in der Weiterentwicklung ein, während die aktuelle Version als Produkt genutzt wird. Alle Phasen des statischen Phasenmodells können immer wieder durchlaufen werden. Evolutionäres Prototyping kann exploratives und experimentelles Prototyping einschließen, jedoch wird die Software von Anfang an so implementiert, daß zu keinem Zeitpunkt ein "Wegwerfen" notwendig wird.

Im Prototyping wird zusätzlich eine horizontale und eine vertikale Vorgehensweise unterschieden:
horizontal: man versucht den gesamten Funktionsumfang eines Systems abzudecken, wobei Funktionen nur ansatzweise implementiert werden. Das Vorgehen entspricht einem Top-Down Entwurf.
vertikal: es wird nur ein kleiner Ausschnitt der Funktionalität realisiert, der aber relativ vollständig ist. Das Vorgehen entspricht einem Bottom-Up Entwurf.

1.4.2 Benutzerbeteiligung im Prototyping-Prozeß

Prototyping bietet die Möglichkeit, bereits früh im Entwicklungsprozeß Bewertungsphasen einzuführen, in denen die in Kapitel 1.1 geforderte Benutzerbeteiligung gut verwirklicht werden kann. Im Entwicklungsprozeß entsteht dann eine Struktur, die sowohl Benutzer und Computersystem (Systemergonomie) als auch die aktive Rolle des Systementwicklers (Software Engineering) berücksichtigen muß. In dieser Struktur sind die in Bild 1.3 dargestellten Schnittstellen bzw. Kommunikationswege vorhanden. Der Benutzer hat ein mentales Modell von seiner Aufgabe, das er in Form einer Aufgabenbeschreibung soweit wie möglich formulieren kann. Auf eine Arbeitsaufgabe im medizinischen Bereich bezogen können hier auch die Randbedingungen des medizinischen Umfelds berücksichtigt werden. Der Systementwickler versucht, unter Berücksichtigung dieser Aufgabenbeschreibung eine Aufgabenanalyse durchzuführen.

Die Relation zwischen Systementwickler und Computersystem ist durch den Einsatz von Programmiermethoden und Werkzeugen charakterisiert. Das Erlernen und Bedienen eines Computersystems durch den Benutzer repräsentiert die Relation Benutzer-Computersystem. Die Benutzerbeteiligung während der Systementwicklung spiegelt sich in der Beziehung zwischen Systementwickler und Benutzer wider. In dieser Relation kommen auch Werkzeuge zur Gestaltung mit Benutzerbeteiligung und die heuristische Bewertung zum Tragen. Auf diese Weise veranschaulicht Bild 1.3 die Verknüpfung von Aspekten aus den Bereichen Systemergonomie und Software-Engineering, die in den Prozeß des evolutionären Prototyping von Benutzerschnittstellen einfließen.

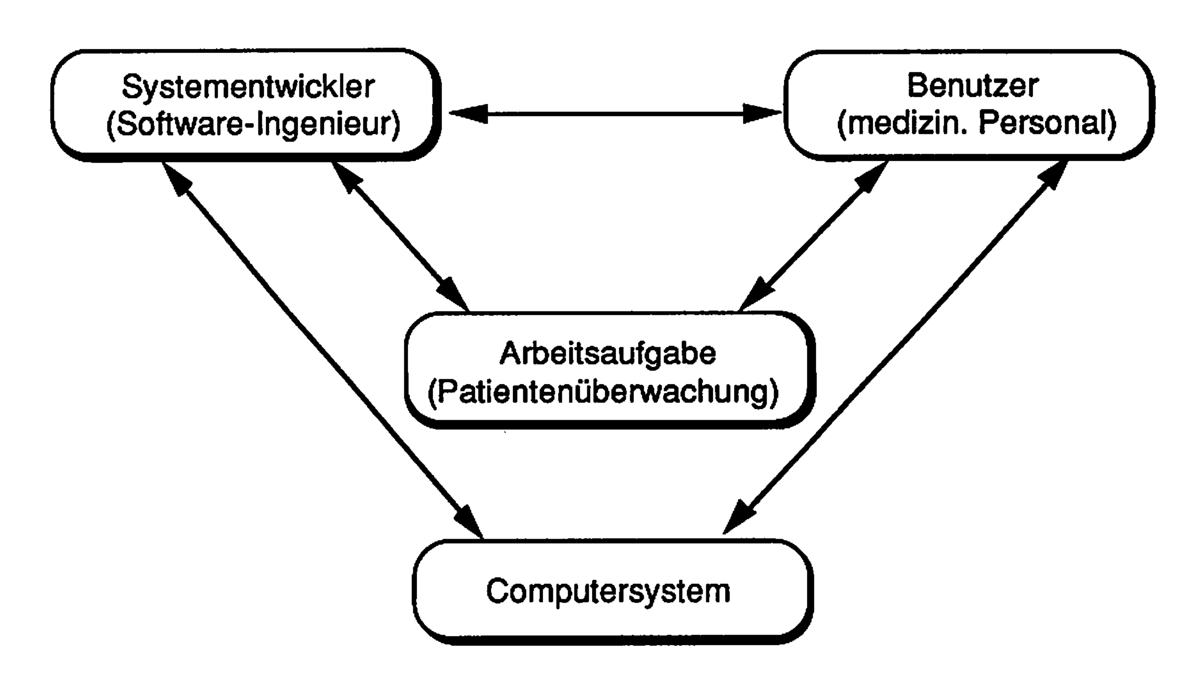


Bild 1.3: Struktur von Relationen bei Prototyping mit Benutzerbeteiligung.

2. Evolutionäres Prototyping

In dieser Arbeit wird der Ansatz des evolutionären Prototyping verfolgt, weil diese Vorgehensweise der Realität von fortwährend veränderten Anforderungen bei der Entwicklung einer Benutzerschnittstelle für ein medizinisches Informationssystem am besten gerecht wird. Es ist nicht möglich, für diese Systeme zu Beginn des Projekts eine vollständige Spezifikation zu erstellen.

Die Entwicklung der interaktiven Benutzerschnittstelle soll dabei ausgehend von einem systemergonomischen Ansatz die perzeptiven und kognitiven Fähigkeiten des Menschen berücksichtigen. Diese Fähigkeiten können nicht durch exakte Modelle beschrieben werden, so daß ein iteratives Vorgehen derzeit der einzig gangbare Weg ist (Rau 1982). Die in der Systemergonomie geforderte Phase der Bewertung durch den Benutzer kann so in jeden Entwicklungszyklus miteinbezogen werden. Dadurch können dann auch die besonderen Merkmale des medizinischen Umfelds bei der Systemgestaltung besser berücksichtigt werden.

2.1 Vorteile und Risiken des evolutionären Prototyping

Die Vorgehensweise des evolutionären Prototyping verspricht viele Vorteile, sie kann jedoch nicht als neuartiges "Allheilmittel" angesehen werden, das alle vorherigen Erkenntnisse des Software Engineering ersetzt. Daher wird der Versuch gemacht, die Vorteile und Risiken des Prototyping gegenüberzustellen. Aufgrund eigener Analysen und aus Literaturhinweisen (Neuser 1987; Spitta 1989) werden zunächst die Vorteile aufgeführt:

1. Frühe Verfügbarkeit einer ersten Version als "vorzeigbarer" Meilenstein.
2. Die Unterstützung der (wenn auch vorläufigen) Spezifikation und deren Verfeinerung durch eine "vorzeigbare" Version.
3. Die Kommunikation mit der Benutzergruppe ermöglicht eine Aufstellung der Benutzeranforderungen, deren stetige Anpassung und exakte Erfüllung.
4. Die Beteiligung der Benutzer während der Entwicklung fördert die Akzeptanz bei der Einführung des Systems.

Die positiven Wirkungen der Benutzerbeteiligung während des Entwicklungsprozesses sind in Bild 2.1 zusammengefaßt (nach Mumford und Welter 1984).

Entscheidung, ein neues System einzuführen.
Bildung einer interdisziplinären Gruppe
für Projektplanung und Systemgestaltung.

Die Beteiligung der Benutzer im Gestaltungsprozeß:
• erleichtert die Lösung von Zielkonflikten,
• erhöht das Vertrauen der Benutzer in die Gestaltung,
• vergrößert das Verständnis auf beiden Seiten;
• vergrößert die Bereitschaft Probleme mitzuteilen
und gemeinsam zu lösen.

Größeres Verantwortungsgefühl für das System bei
der Neueinführung.
Zukünftige Systeme können auf der Basis größerer
Qualifikation entwickelt werden, da Benutzer und
Systemtechniker über mehr Wissen verfügen.

Bild 2.1: Wirkung der Benutzerbeteiligung bei der Systementwicklung (nach Mumford und Welter 1984).

Die Probleme und Risiken des Prototyping sind:

1. Es kann die Tendenz entstehen, die Anforderungsanalyse zu vernachlässigen und stattdessen ungezielt zu "probieren". Gestaltungsprobleme sollen nicht durch Prototyping gelöst werden, wenn eine Aufgabenanalyse schneller zur Lösung führt..
2. Die Anzahl von Programmfehlern steigt durch jede Änderung erneut an (degenerierender Kode).
3. Frühe Versionen sehen oft bereits fertig aus; dies kann die Motivation zur notwendigen Detailarbeit verringern oder zur Fehleinschätzung der Projektdauer führen.
4. Aufgrund von häufig veränderten Systemanforderungen kann die Motivation der Entwickler für die Implementierung beeinträchtigt werden.
5. Das Projektmanagement (externe Kontrolle, Zeit-Planung, ..) ist schwierig handhabbar.

Die genannten Risiken des Prototyping können durch eine sorgfältige Aufgabenanalyse allein nicht ausgeschlossen werden. In dieser Dissertation wird daher das Ziel verfolgt, eine Entwicklungsumgebung zu entwerfen, die die genannte Vorgehensweise unterstützt und dabei gleichzeitig den Risiken entgegenwirkt.

Der Schwerpunkt dieser Arbeit liegt zunächst auf der Erarbeitung der Anforderungen an eine Prototyping Entwicklungsumgebung und der Realisierung von geeigneten

Software-Werkzeugen. Diese Software-Werkzeuge werden so konzipiert, daß eine kontinuierliche Benutzerbeteiligung während des Entwicklungsprozesses vereinfacht wird. Da dieser Prozeß ein besonderes Projektmanagement erfordert, wird im nächsten Abschnitt auf das Projektmanagement im Prototyping eingegangen, bevor das Konzept für eine Entwicklungsumgebung in Abschnitt 2.3 entworfen wird,

2.2 Aspekte des Projektmangement im Prototyping-Prozeß

Die Vorgehensweise des evolutionären Prototyping erfordert ein tiefgreifendes Umdenken bei der Projektgestaltung. Die Bereitwilligkeit für eine partnerschaftliche Zusammenarbeit zwischen Entwickler und Benutzer ist abhängig sowohl von menschlichen Grundhaltungen der Beteiligten als auch von organisatorischen Bedingungen.

2.2.1 Organisation des Projektablaufs

Die Überwachung und Kontrolle des Entwicklungsablaufs unterscheidet sich erheblich von der konventionellen Projektkontrolle, die sich am Pflichtenheft und dem klassischen Phasenmodell orientiert. Im Prototyping ist die Anzahl notwendiger Iterationen im Entwicklungsprozeß zunächst unbestimmt. Daher muß das Gesamtprojekt in einzelne überschaubare Abschnitte unterteilt werden, die durch "Meilensteine" gekennzeichnet werden. Diese Meilensteine können aus den Ergebnissen der Anforderungsananlyse abgeleitet werden. Da evolutionäres Prototyping als ein Prozeß ohne definiertes Ende zu verstehen ist, muß ein Projekt irgendwann zumindest vorübergehend als "fertig" erklärt werden. Hierbei orientiert man sich an der dem Projekt zugrundeliegenden Anforderungsanalyse: ein Projekt wird als "vorübergehend fertig" betrachtet, solange keine neuen Anforderungen bekannt sind und alle bekannten Anforderungen erfüllt sind. Sobald neue Anforderungen an eine Version gestellt werden, beginnt ein neuer Iterationsschritt.

Im praktischen Ablauf eines Projekts bedeutet das typischerweise, daß in regelmäßigen Projektbesprechungen der Stand und die nächsten Schritte des Projekts protokolliert und zeitlich festgelegt werden, um so einen systematischen Informationsaustausch unter den Beteiligten zu fördern.

2.2.2 Benutzerbeteiligung

Eine wesentliche Zielsetzung im Prototyping ist die Verbesserung der Kommunikation zwischen Systementwicklern und Benutzern. Diese Kommunikation beruht auf dem Austausch von Nachrichten, die jeweils verschiedene Ebenen enthalten (Bild 2.2).

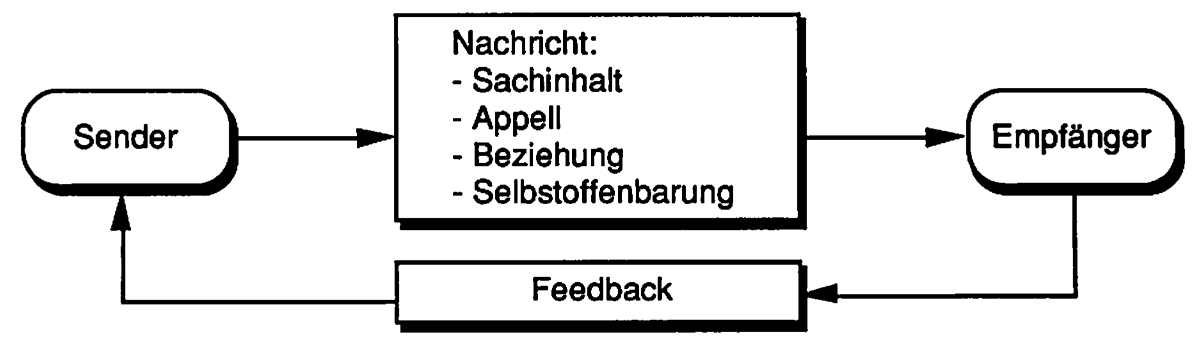


Bild 2.2: Kommunikationsmodell nach Fittkau (1980).

Die Annahme, daß eine Projektarbeit nur auf der Sachebene stattfinden könnte, hat in der Vergangenheit Projektfehlschläge in Millionenhöhe verursacht (Hofstetter 1985). Die verschiedenen Verhaltensweisen hängen oft davon ab, ob der Benutzer durch das Projekt eine Arbeitserleichterung, eine Qualitätsverbesserung der Arbeit oder Prestigegewinn erwartet (Spitta 1989). Bei Berücksichtigung der verschiedenen Kommunikationebenen können folgende Typen von Benutzer unterschieden werden:

- der Benutzer versteht sich als Adressat von Leistungsanforderungen und reagiert nur auf Appell (passiv-neutral),
- der Benutzer ist aufgrund einer positiven Beziehung an wechselseitigem Austausch von Information und Appellen interessiert (aktiver Partner),
- der Benutzer beteiligt sich hauptsächlich korrektiv, d.h. mit dem Schwerpunkt auf dem Feedback (aktiv-kritischer Partner),
- der Benutzer hat eine negative Beziehung zum Entwickler und versucht das Projekt zu behindern (aktiv-feindlicher Gegner).

Die Auswahl von aktiven Partnern für die Zusammenarbeit ist für die Projektentwicklung besonders positiv. Wenn Benutzer nur als Vertreter per Amt und Funktion in die Mitarbeit einbezogen werden, so ist mit deren Desinteresse zu rechnen. Die Erfahrung auf dem interdisziplinären Gebiet der Medizintechnik hat gezeigt, daß eine kleine Anzahl von Experten auf Dauer nicht ausreichend ist, selbst wenn die kleine Gruppe hochmotiviert ist. Oft sind diese Experten die einzigen aus der Benutzergruppe mit technischem Interesse und selbst aber nur gelegentlich Benutzer des neuen Systems. Es sollen jedoch insbesondere die "typischen" späteren Benutzer des neuen Systems an der Entwicklung beteiligt sein.

Ein charakteristisches Problem des Prototyping besteht darin, die Benutzer über einen längeren Zeitraum zur Zusammenarbeit zu motivieren (Williges 1987). Um die Motivation aufrecht zu erhalten, können die Methoden der heuristischen Evaluierung variiert werden. Nach Dick und Carey (1978) werden Einzel-, Kleingruppen- und Feldtests unterschieden. In Anlehnung daran wurden folgende Verfahren durchgeführt:

1. Persönliches Gespräch mit einem Benutzer; zunächst informell, später mit Diskussion direkt am Bildschirm. Hierbei besteht die Möglichkeit, daß der Systementwickler die Bedingungen aus dem Arbeitsumfeld des Benutzers besser kennenlernt und umgekehrt der Benutzer mehr von den technischen Möglichkeiten der Systemgestaltung kennenlernt.
2. Diskussion innerhalb einer kleinen Gruppe mit gleicher Anzahl von Entwicklern und Benutzern. Dabei sollen gemeinsame Zielsetzungen erarbeitet und festgehalten werden, um sie zu einem späteren Zeitpunkt anhand neuer alternativer Versionen aufgreifen und weiterdiskutieren zu können.
3. Fragebogen und strukturierte Interviews eignen sich besonders gut zur Durchführung direkt am Arbeitsplatz des Benutzers.
4. Labortest. Eine vollständige oder in einem Teilbereich vollständige Implementation wird vom Benutzer analog zu einer realen Aufgabenstellung im Labor bedient. Dabei kann die objektive Leistung (Performance) und die subjektive Einschätzung des Benutzers bestimmt werden.
5. Feldtest. Eine funktional vollständige Version des Systems wird im Arbeitsumfeld eingesetzt.

Die Punkte 1 bis 4 können häufig und auch mit unterschiedlich vielen Personen durchgeführt werden. Mit der Zeit verschiebt sich der Schwerpunkt von einem Informationsaustausch ("was soll gemacht werden") zu einem Meinungsaustausch ("wie soll es gemacht werden").

2.2.3 Berücksichtigung von Interessenkonflikten

Die Zusammenarbeit von Systementwickler und Benutzern erfordert eine positive Grundeinstellung zum Projektablauf. Wenn der Systementwickler die Methodik des Prototyping nicht bewußt unterstützt, könnte er möglicherweise auf eine schnelle Festschreibung einer Version drängen, weil jede Änderung gleichzeitig zusätzliche Arbeit bedeutet. Aber auch die entgegengesetzte Position ist kontraproduktiv: die schnelle Änderbarkeit in einer leistungsfähigen Entwicklungsumgebung kann zu einer unreflektierten "Prototyping-Akrobatik" (Case 1985) führen, die möglichst alle Variationen "ausprobieren" will. Dabei entfernt sich der Entwickler zunehmend von den Bedürfnissen des Benutzers, so daß ein Projekt nie die Reife für einen Feldtest erreicht.

Auf der anderen Seite kann der Benutzer ein Projekt behindern, indem er unrealistische Anforderungen stellt, die der Prototyp (noch) nicht erfüllen kann; oder er entwickelt aus Mangel an ihm zur Verfügung stehender Zeit kaum Alternativen. Eine Vielzahl unterschiedlicher Interessengruppen behindert die Effizienz von Entscheidungsprozessen und erzeugt Zeitverlust, Überforderung und inkonsistente Konzepte (Neuser 1987). Ein strukturiertes Gesamtkonzept sollte daher von Beginn an in den Händen der Systementwickler liegen.

Abschließend läßt sich festhalten, daß die Zusammenarbeit bei Systementwicklern und beteiligten Benutzern echte Kompromißbereitschaft bei der gemeinsam vereinbarten Zielsetzung erfordert, damit die Kooperation nicht nur als notwendig erachtet, sondern auch tatsächlich Bestandteil der Handlung wird.

2.3 Festlegung des Konzepts für eine Entwicklungsumgebung

Der Einsatz einer Software-Entwicklungsumgebung zielt auf die Verbesserung von Qualität und die Verkürzung der Entwicklungsdauer. Durch die Iterationen im Prototyping-Prozeß kann ohne Hilfsmittel die Entwicklungsdauer stark ansteigen. Hinzu kommt ein schwieriges Problem konventioneller Programmier-Methoden: der erzeugte Kode degeneriert durch die häufig durchgeführten Änderungen (Bild 2.3).

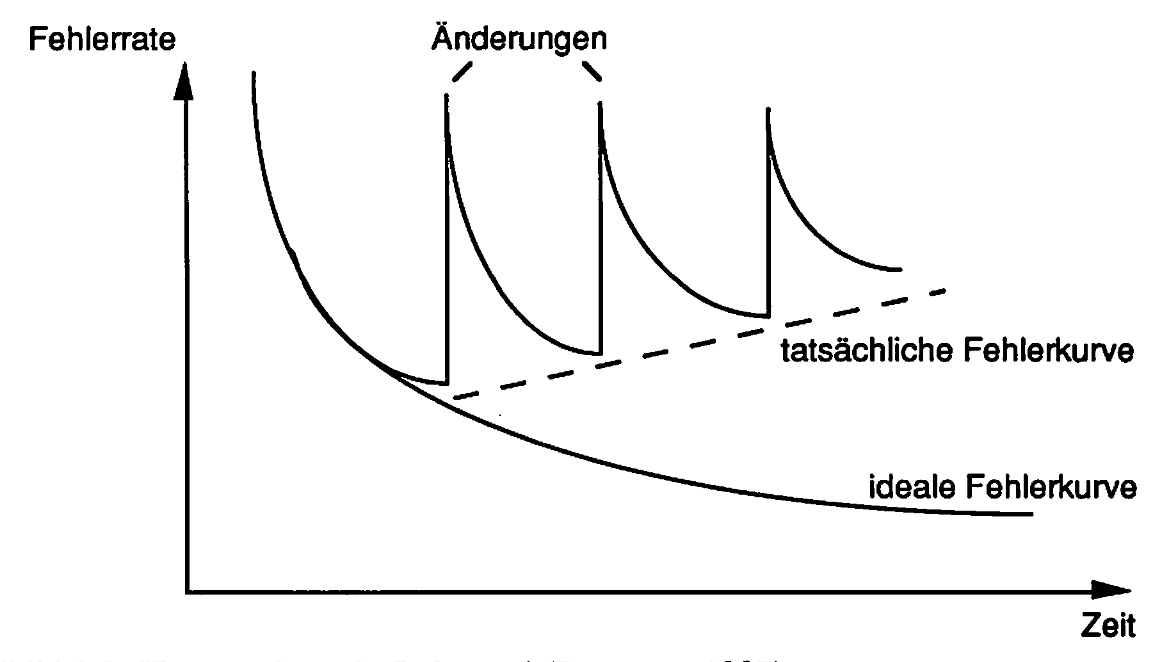


Bild 2.3: "Degenerierende Software" (Pressman 1987).

Durch eine Bibliothek mit wiederverwendbaren Softwaremoduln kann dieser degenerative Prozeß verhindert werden. Eine solche Bibliothek ist dem Ansatz einer "Bausteinbank" aus der strukturierten Programmierung verwandt. Aber erst objektorientierte Sprachen wie Smalltalk, C++ oder Objective-C erfüllten die entscheidenden Vorraussetzungen für die Realisierung von anwendungsneutralen, wiederverwendbaren Moduln (Cox 1986, Endres 1988). Die Kennzeichen einer solchen Realisierung sind eine in sich abgeschlossene Funktionalität von Objekten nach dem Geheimnisprinzip, Informationsaustausch durch Versenden von Nachrichten und Vererbungsmechanismen (vgl. Kap. 3.1). Cox (1986) nennt diese wiederverwendbaren Moduln "Software ICs". Der Aufwand einer Programmentwicklung verteilt sich dann auf den "IC-Hersteller" und den "IC-Anwender", so daß eine Erleichterung und Zeitersparnis für den IC-Anwender erreicht wird. Der einmal in eine IC-Herstellung investierte Arbeitsaufwand kann vom Anwender vielfach genutzt werden, wenn ein "IC" genügend flexibel entworfen wurde.

Eine Bibliothek, die über vorgefertigte interaktive Elemente einer Benutzerschnittstelle verfügt, verspricht eine starke Produktivitätssteigerung, da in interaktiven Systemen mehr als die Hälfte des Entwicklungsaufwands in die Benutzerschnittstelle fließt (Rowe und Shoens 1983). Je umfangreicher diese Bibliothek ist, desto weniger Bausteine müssen während eines Projekts neu erstellt werden (Bild 2.4).

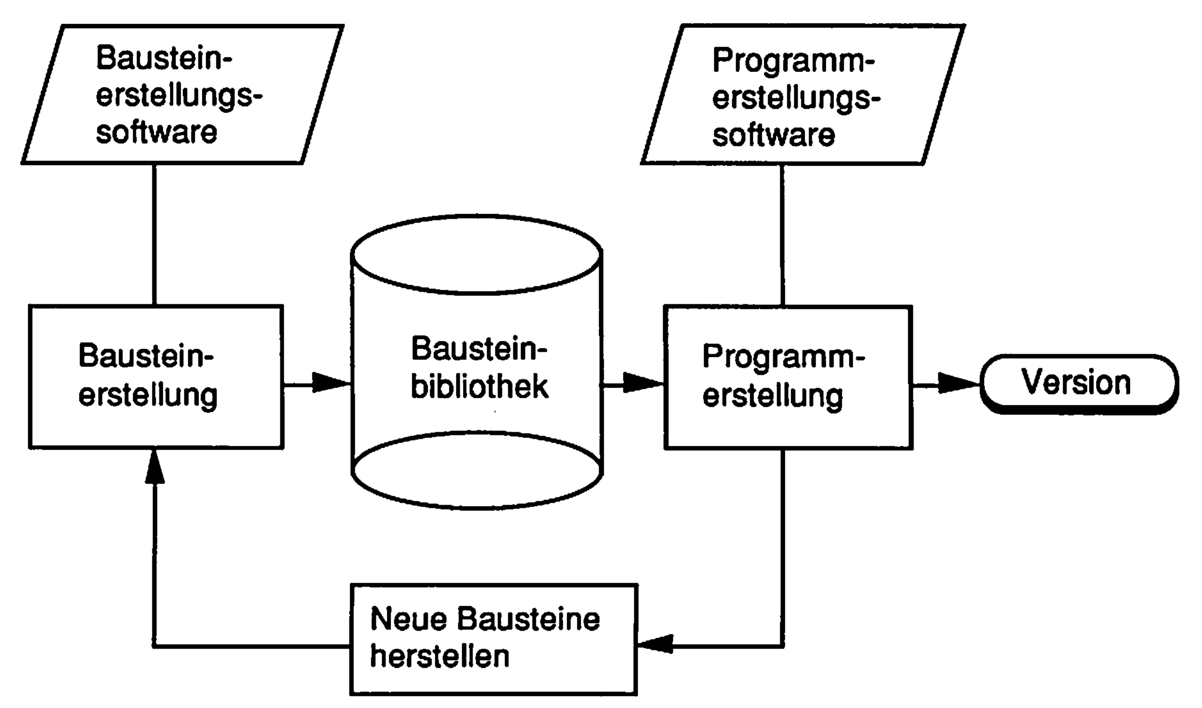


Bild 2.4: Programmentwicklung mit Bausteinbibliothek (Platz 1983).

In den letzten Jahren wird verstärkt der Ansatz verfolgt, die Software der Benutzerschnittstelle unabhängig von der Applikationssoftware zu entwickeln. Die Entwicklung der Benutzerschnittstelle kann dabei durch sog. "user interface management systems" (UIMS) unterstützt werden (Pfaff 1985), die die Beschreibung der Ein-/Ausgabe und/oder der Dialogführung in interaktiven Systemen vereinfachen.

Die Anwendung von Prototyping-Verfahren unter Einsatz eines UIMS wird als besonders effektive Methode angesehen (Wilson und Rosenberg 1988), es wurden bislang jedoch selten UIMSs eingesetzt. Dies liegt z.T. daran, daß UIMSs bisher nur für Teilaspekte einer Entwicklung ausgelegt waren. Perlman (1988) und Farooq (1988) regen daher an, UIMSs zu einer integrierten Entwicklungsumgebung mit Werkzeugen zur Gestaltung und Bewertung auszubauen.

In Bild 2.5 sind einige aus der Literatur bekannte UIMSs und "Interface Toolkits" aufgeführt. Die Integration von Editoren ist erst in wenigen Entwicklungsumgebungen berücksichtigt, Ansätze zur Bewertung sind bisher überhaupt nicht vorhanden.

	SYNGRAPH Olsen 1983	Alberta UIMS Green 1985	COUSIN Hayes et al. 1985	THESEUS Hübner 1987	GW-UIMS Silbert et al. 1988	Peridot Meyers 1987	HP-Widgets HP 1989	MacApp Wetmore 1988	VAPS VPI 1988	NextStep Thompson 1989
Interaktion:										
direkt-manipulativ				●	●	●	●	●	●	●
Kommando-orientiert	●	●	●							
Interaktionsmodi:										
Mauseingabe				●	●	●	●	●	●	●
Berühreingabe										
Spracheingabe										
Sprachausgabe										●
angebundene/integrierte Editoren								●	●	●

Bild 2.5: Übersicht bekannter UIMSs (mit Angabe der Literaturquellen).

Die beschriebenen Vorteile und notwendigen Anforderungen bilden die Basis zur Formulierung der wichtigsten Zielsetzungen der vorliegenden Dissertation, die im folgenden thematisch skizziert werden.

Als Kern einer Entwicklungsumgebung wird ein Interfacebaukasten entwickelt und vorgestellt, der im Sinne eines UIMS die Methodik des evolutionären Prototyping unterstützt. Die wiederverwendbaren Softwarebausteine des Interfacebaukastens bilden die Grundlage für einen schnellen Entwurf von Benutzerschnittstellen mit Benutzerpartizipation. Diese Bausteine unterstützen den Systementwickler derart, daß sowohl im Entwurf als auch während der Implementation eine aufgabenbezogene, gestalterische Sichtweise beibehalten werden kann, d.h. der Schwerpunkt seiner Arbeit liegt auf den Objekten einer Benutzerschnittstelle und deren Attributen, nicht aber auf Graphikprogrammierung oder Programmroutinen für Abfrage und Auswertung von Interaktionen.

Zusätzlich zu diesem Baukasten werden Gestaltungswerkzeuge geschaffen, die gemeinsam mit dem UIMS aufeinander abgestimmt in einer Entwicklungsumgebung integriert sind. Diese Werkzeuge basieren auf Kenntnissen von Perzeption und Kognition des Menschen, d.h. für eine Anpassung eines technischen Systems an den Mensch als Benutzer ist es notwendig, vom Mensch als Systemkomponente soviel wie möglich zu wissen und bei der Gestaltung geeignet zu berücksichtigen (Rau 1979). Damit wird erstmals eine Werkzeugumgebung so aufgebaut, daß für eine optimale Informationsdarstellung und Interaktion die Fähigkeiten und Grenzen des Menschen geeignet berücksichtigt werden können. Bei der Informationsaufnahme und Verarbeitung spielt die Unterscheidung zwischen verbaler (Text, Sprache) und räumlicher Information (Graphik) eine herausragende Rolle. Aus der Hirnforschung ist bekannt, daß jede dieser beiden Informationstypen jeweils einer der beiden Gehirnhälften zugeordnet werden kann; die Verarbeitung von verbaler und räumlicher Information verläuft daher getrennt (z.T. parallel) und mit unterschiedlichen Eigenschaften. Die hier konzipierte Entwicklungsumgebung enthält jeweils 2 Werkzeuge mit Bezug auf verbale Information und auf räumliche Information. Diese Werkzeuge bilden zusammen mit dem Interfacebaukasten eine Entwicklungsumgebung für interaktive Informationssysteme. Bild 2.6 zeigt den Interfacebaukasten als Kern der Entwicklungsumgebung, um den Editoren für den Schnittstellenentwurf, die Farbgestaltung, Spracheingabe und akustische Meldungen angeordnet sind.

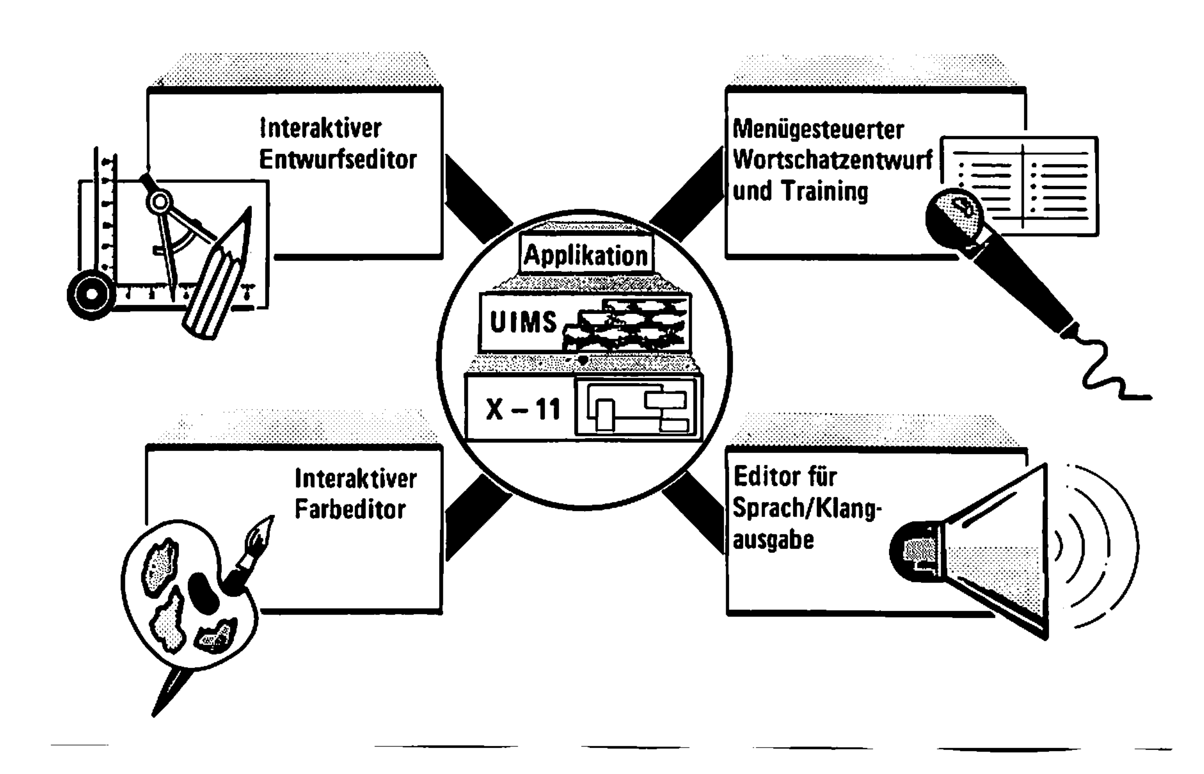


Bild 2.6: Entwicklungsumgebung für multimodale Benutzerschnittstellen (Langen 1990).

Die Integration von Bewertungsmethoden in eine Entwicklungsumgebung wird als eine der vielversprechensten Verbesserungen (vgl. Bild 2.5) zukünftiger Prototyping-Systeme angesehen (Lowgren 1988, Perlman 1988, Farooq 1988). Die Berücksichtigung von Ansätzen zur Bewertung von Benutzerschnittstellen wurde daher in den verschiedenen Werkzeugen dieser Entwicklungsumgebung integriert. Dies umfaßt die Bewertung:

- der geometrischen Anordnung von Objekten am Bildschirm,
- der farblichen Gestaltung,
- von Piktogrammen,
- von trainierten Wortschätzen zur Spracherkennung,
- von akustischen Meldungen.

Im folgenden Kapitel wird zunächst der Interfacebaukasten vorgestellt. Danach folgt die Beschreibung der Werkzeuge, aufgeteilt in die Gruppe von Werkzeugen in Bezug auf visuelle Informationsdarstellung (Kapitel 4) und der Gruppe bezüglich multimodaler Interaktion (Kapitel 5). In einer exemplarischen Gestaltung eines sehr komplexen Informationssystems (Kapitel 6) wird dann von den Werkzeugen Gebrauch gemacht und die Leistungsfähigkeit des erarbeiteten Gesamtkonzeptes gezeigt.

3 Entwicklung eines objektorientierten Baukastensystems

Das Baukastensystem bildet die softwaretechnische Grundlage für den iterativen Prozeß des evolutionären Prototypings. Für die Festlegung des Lösungskonzepts des Baukastensystems wird ein Modell für direkt-manipulative Benutzerschnittstellen aufgestellt und dann darauf aufbauend die Objektstruktur des Baukastensystems bestimmt.

3.1 Beschreibung des Lösungskonzepts für den Interfacebaukasten

3.1.1 Zielsetzung von Management Systemen für Benutzerschnittstellen (UIMS)

In den letzten Jahren wurden zahlreiche Systeme entworfen, die den Entwurf von Benutzerschnittstellen unterstützen sollen. Für diese Systeme hat sich der Begriff "user interface mangagement system" (UIMS) in der Literatur etabliert. Eine weitgehend akzeptierte Definition eines UIMS formulierte Norman (1986):

> "A UIMS provides a way for a designer to specify the interface in a high-level language. The UIMS then translates that specification into a working interface, managing both the details of the display and its associated input and output and also the interaction with the rest of the program. UIMS systems allow the generation of high-quality interfaces with much less effort than programming the interfaces directly"

Löwgren (1988) faßt die Charakteristika eines UIMS konkreter:
- es ermöglicht konsistente Benutzerschnittstellen, sowohl innerhalb einer Applikation (Konsistenz von Teilschnittstellen und deren Elementen) als auch zwischen zusammengehörigen Applikationen (einheitliches "look and feel", "house style"),
- es erleichtert die Modifikation der Benutzerschnittstelle,
- es fördert die Entwicklung und Nutzung wiederverwendbarer Software-Komponenten,
- es hebt sich von der konventionellen Programmierung ab,
- es unterstützt die Entwicklung leicht lernbarer und leicht benutzbarer Schnittstellen.

Der hier vorgestellte Interfacebaukasten für interaktive Schnittstellen (IBIS) ist so aufgebaut, daß bereits durch die einzelnen Bausteine Prinzipien ergonomischer Gestaltung implizit berücksichtigt werden (sog. "built-in expertise"). So kann z.B. die Farbkodierung als Mittel der Benutzerführung fest integriert werden.

Im einzelnen wurden folgende Ziele gesetzt:

- **Hierarchische Strukturierung**
  Die von einer Aufgabe abzuleitende Aufgabenstruktur soll direkt in die Implementation einer Benutzerschnittstelle übertragen werden können. Einer Teilaufgabe soll eine Teilschnittstelle zugeordnet werden, die eine Gruppe von Bedienelementen und dazugehörigen Interaktionssequenzen zusammenfaßt.
- **Multimodale Interaktion**
  Das Prinzip der "direkten Manipulation" soll auf multimodale Interaktion (Berühreingabe, Mauszeiger, Sprachein- und Sprachausgabe) erweitert werden.
- **Konsistente Benutzerschnittstellen**
  Konsistenz von Präsentation und Interaktion der Benutzerschnittstelle. In jedem Baustein sind bereits Mechanismen und Voreinstellungen für Darstellung und elementare Interaktion (Basis-Dialog) bereitzustellen.
- **Aufgabenbezogene Quellkodierung**
  Eine Implementierungssprache soll nicht nur der Kodierung, sondern gleichzeitig auch der Spezifikation dienen. Die Semantik dieser Sprache soll sich direkt auf die für den Entwickler relevanten, anwendungsspezifischen Attribute beziehen, so daß durch die größere Transparenz die Modifizierbarkeit und Wartbarkeit großer Softwaresysteme erleichtert wird.

3.1.2 Objektorientierte Programmierung

Die im letzten Abschnitt gesetzten Ziele an ein Baukastensystem können besonders gut durch einen objektorientierten Ansatz erfüllt werden. Objektorientierte Programmierung (OOP) unterscheidet sich von der herkömmlichen Art der Programmierung dadurch, daß nicht mehr der (prozedurale) Programmierschritt, sondern ein Objekt im Mittelpunkt steht. Ein Objekt ist ein weitgehend unabhängiges Gebilde, dessen innerer Aufbau nicht bekannt sein muß, um es zu verwenden. Ein objektorientierter Programmierer betrachtet die Welt als System miteinander kommunizierender Objekte. Die Kontrollstruktur eines Programms wird nicht mehr durch Schleifen oder Verzweigungen bestimmt, sondern durch das Versenden und Empfangen von Nachrichten. Die Betrachtungsweise verändert sich dahin, daß man benennt "WAS" gemacht werden soll, nicht "WIE" etwas gemacht werden muß. OOP ist nicht so sehr eine neuartige Kodierungsform, sondern vielmehr eine besondere Technik, Programmkode aus einer neuen Sichtweise heraus zu verpacken. Drei Konzepte aus dem Software Engineering werden dabei konsequent verwirklicht: Abstraktion, Geheimnisprinzip und Modularisierung

Die zentralen Begriffe der OOP werden nachfolgend kurz erläutert;

Objekt: Träger von privaten Daten und auf diesen Daten operierender Methoden. Methoden werden aufgrund von an das Objekt gesendeten Nachrichten ausgelöst.

Verkapselung: Objektdaten sind nach Außen hin geschützt (privat). Sie werden nur mittelbar über Nachrichten angetastet.

Nachrichten: Objekte kommunizieren über Nachrichten, die im Objekt festgelegte Methoden auslösen. Im Unterschied zu Funktionsaufrufen ist die Reaktion auf eine Nachricht abhängig vom empfangenden Objekt, d.h. gleiche Nachrichtennamen können verschiedenartige (verkapselte) Algorithmen aktivieren (Polymorphie). Dadurch kann der innere Zustand verändert werden, ein Rückgabewert erzeugt oder eine weitere Nachricht verschickt werden.

Methode: Beschreibung eines Algorithmus innerhalb des Objekts, der als Reaktion auf eine bestimmte Nachricht abgearbeitet wird.

Objekt-Klasse: Objekte mit gleichen Eigenschaften werden in einer Klasse zusammengefaßt. So ist z.B. "Fahrrad" eine Objekt-Instanz der Klasse "Fortbewegungsmittel", d.h. ein Fahrrad hat wesentliche Merkmale mit anderen Fortbewegungsmitteln gemeinsam.

Vererbung: Als Mitglied einer Klasse (Instanz) beginnt ein neues Objekt nicht als unbeschriebenes Blatt. Die Eigenschaften der Klasse und deren Oberklassen sind automatisch verfügbar.

Der Interfacebaukasten für interaktive Schnittstellen (IBIS) wurde in der Sprache "Objective-C" implementiert; es handelt sich bei dieser Sprache um eine objektorientierte Erweiterung der Sprache "C". Die Objekte des Baukastensystems besitzen einen privaten Datensatz (Instanzvariablen), bestehend aus einem modifizierbaren Teil und einem festgeschriebenen Anteil. Der festgeschriebene Anteil bewirkt, daß bestimmte ergonomische Prinzipien (z.B. Art der Benutzerführung) einheitlich definiert werden und dadurch die Konsistenz einer Benutzerschnittstelle unterstützt wird. Der modifizierbare Datenanteil besitzt stets eine Voreinstellung ("sinnvoll gewählter Defaultwert"), die den ersten Entwurf stark vereinfacht; diese Daten können durch Senden von Nachrichten bei Bedarf applikationsgerecht verändert werden. Da der Entwickler mit vorgegebenen Nachrichten arbeitet, können bestimmte Beziehungen innerhalb der Objektwelt automatisch aufrechterhalten werden (sog. "Constraints"). Der Entwickler beschreibt deklarativ (nicht prozedural) die von ihm gewünschten Elemente der Schnittstelle, das System selbst sorgt dafür, daß dabei vereinbarte Constraints aufrecht erhalten werden (z.B. daß die Beschriftung einer Taste immer zentriert dargestellt wird). Dabei spielen Details der zugrunde liegenden Programmiersprache "C" bzw. des Graphiksystems "X-Window" keine Rolle, so daß die Sichtweise stets auf die Applikation gerichtet bleiben kann.

3.1.3 Modell direkt-manipulativer Benutzerschnittstellen

Für eine differenzierte Betrachtungsweise von Mensch-Maschine Schnittstellen wurde das in der Form eines Schichtenmodells ausgestaltete IFIP-Modell (Bild 3.1) entworfen (Dzida 1983). Dabei wird jede Benutzerschnittstelle in vier Ebenen unterteilt. Die Ein-/Ausgabeebene behandelt Fragen der Informationsdarstellung und der Eingabe von Information durch Stellglieder. Auf der Dialogebene werden geeignete Dialogformen und Bediensequenzen festgelegt. Die Werkzeugebene regelt den Umgang des Benutzers mit Werkzeugen oder Systemfunktionen (z.B. Hilfsfunktionen). Auf der Organisationsebene sind Fragen der Aufgabenverteilung zwischen Mensch und Maschine bzw. innerhalb einer Arbeitsgruppe zu beantworten.

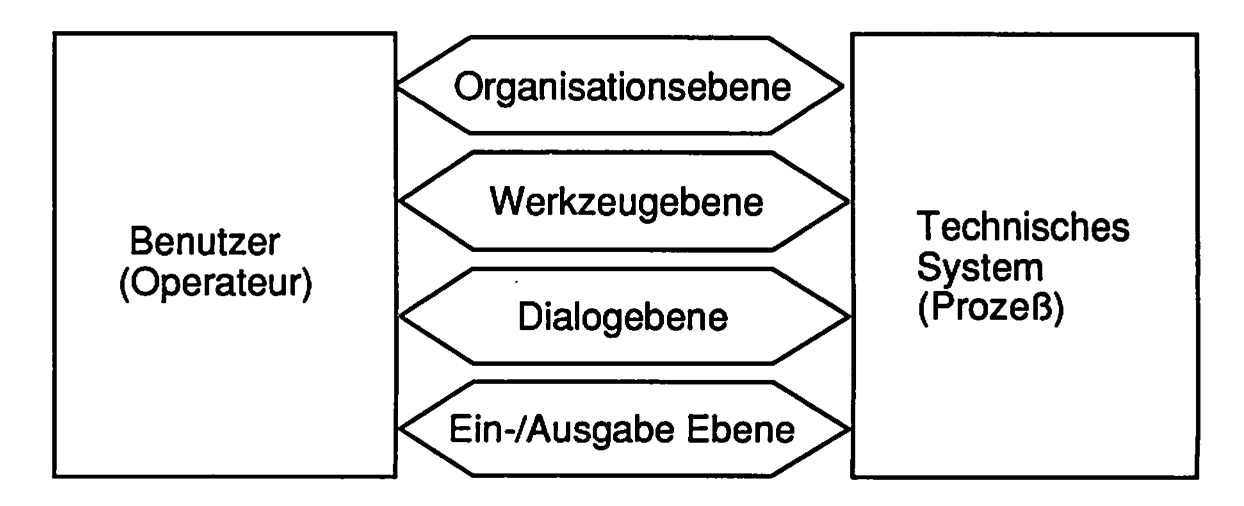


Bild 3.1: IFIP-Schnittstellenmodell.

Auf dem Workshop für "User Interface Management Systems" in Seeheim (Pfaff 1985) wurde ein ähnliches Schichtenmodell entworfen, das die Struktur von Benutzerschnittstellen systematisieren soll. Das "Seeheim-Modell" (Green 1985) für Benutzerschnittstellen (Bild 3.2a) unterscheidet die Präsentationsebene, die Ebene der Dialogführung und die Applikationsebene. Nach zunächst weitgehender Anerkennung des Seeheim-Modells wurde die Kritik geäußert, daß es eher auf Textkommando-Interaktion hin ausgerichtet sei, nicht aber dem Charakter direkt-manipulativer Systeme entspricht (Löwgren 1988). Direkte Manipulation zeichnet sich dadurch aus (Rau 1979; Shneiderman 1982), daß relevante Objekte kontinuierlich auf dem Bildschirm sichtbar sind. Anstatt einer komplexen Syntax werden physische Aktionen (Bewegung, Zeigen) zur Manipulation benutzt. Bei der direkten Manipulation kommuniziert der Benutzer mit einzelnen Objekten und nicht mit dem System als Einheit. Es wird stets eine visuelle Rückkopplung der Manipulation an den Objekten dargeboten. Die Berücksichtigung eines grafischen Feedbacks bei einer Interaktion ändert die Beziehung zwischen den Ebenen des Seeheim-Modells: die Dialogebene kann

nicht mehr als "Vermittler" zwischen Darstellungs- und Applikationsebene gesehen werden (Bild 3.2b).

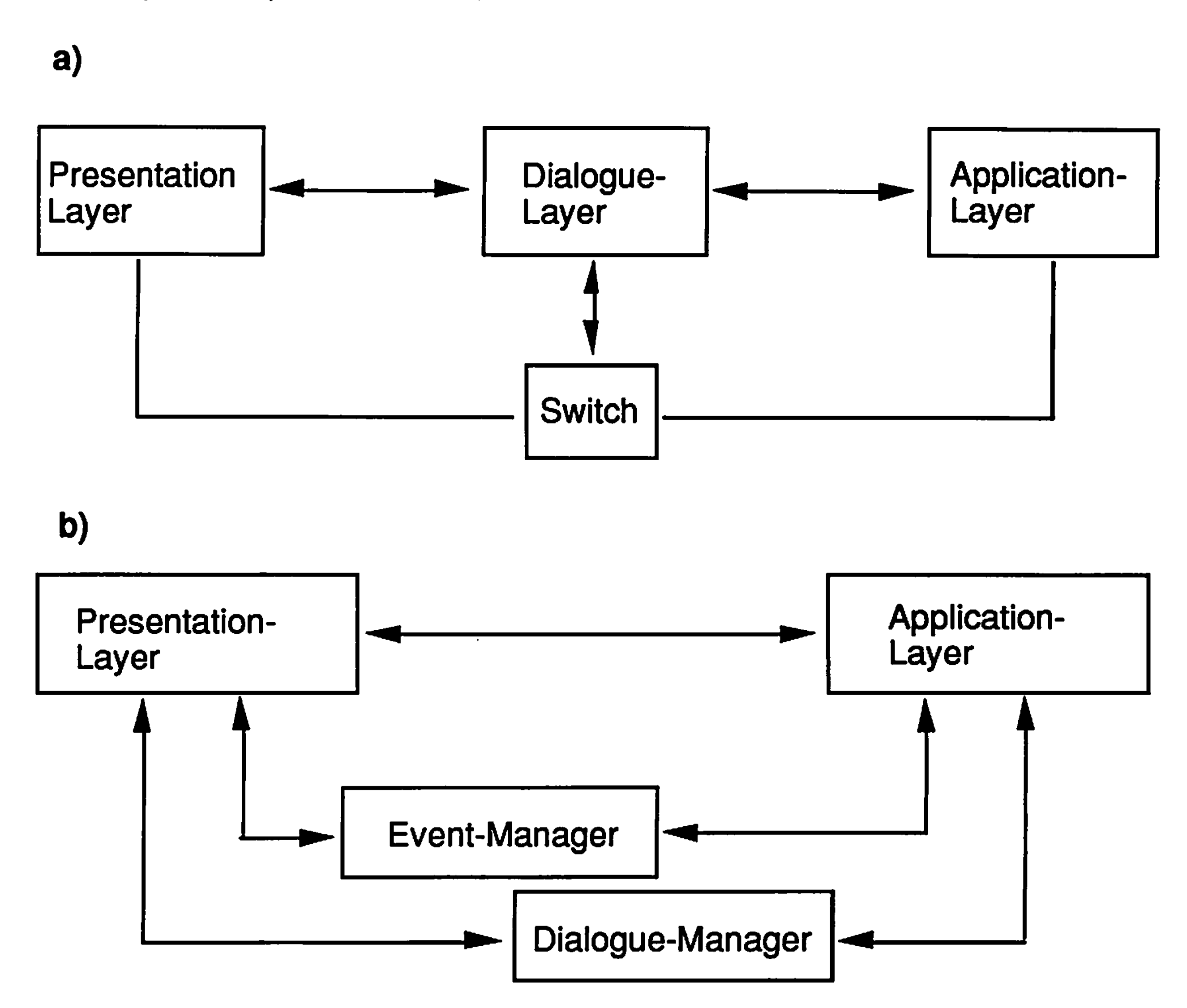


Bild 3.2: a) Seeheim-Modell einer Benutzerschnittstelle.
b) Modell eines direkt-manipulativen Informationssystems.

Im Unterschied zum "Seeheim-Modell" wird für den in dieser Arbeit entwickelten Interfacebaukasten die Dialogkomponente aufgespalten; sie besteht aus einem Interaktionsmanagement für den elementaren Dialog auf der Ebene von Eingabeereignissen eines Objekts und einem komplexen Dialog, der das Zusammenwirken von Interaktionsobjekten untereinander beschreibt. Der elementare Dialog beruht auf dem "Event-Modell" (Green 1984), d.h. der Informationsaustausch erfolgt auf der Basis von Ereignissen, die von Eingabegeräten generiert werden. Diese Ereignisse werden von einem Interaktionsmanagement in die fest definierte Datenstruktur eines "input token" vorverarbeitet und in dieser Form an ein Interaktionsobjekt gesendet, das als Empfänger des "input token" zu einer Reaktion veranlaßt wird. Diese Reaktion ist in

jedem Interaktionsobjekt vollständig verkapselt, eine Rückmeldung (Handshake) an das Interaktionsmanagement ist nicht erforderlich. Dadurch wird erreicht, daß Informationsdarstellung und Interaktion auf der Ebene einzelner Interaktionsobjekte integriert sind. Ein Interaktionsobjekt kann also aufgrund der eigenständigen Interaktivät als "Dialogzelle" (ten Hagen 1985) betrachtet werden. Der komplexe Dialog, d.h. eine die einzelnen Objekte verknüpfende Interaktionssequenz, wird dann von einem Dialogmanager verwaltet.

3.2 Struktur des Interfacebaukastens

Ausgehend von den Anforderungen an direkt-manipulative Benutzerschnittstellen werden folgende drei Gruppen von Objekten unterschieden:
- Interaktionsobjekte,
- Präsentationsobjekte,
- Interaktionsmanagementobjekte (Eventverwaltung).

Interaktionsobjekte reagieren auf Benutzereingaben, während Präsentationsobjekte ausschließlich der Darstellung von Information dienen. Die Klasse der Interaktionsmanagementobjekte nimmt eine synchronisierte Verarbeitung aller Eingabeereignisse zu "input token" vor, die durch Interaktion an unterschiedlichen physikalischen Geräten (multimodal) erzeugt werden.

3.2.1 Interaktionsobjekte

Die Struktur von komplexen Systemen kann durch Abstraktion bzw. Verfeinerung auf verschiedenen Komplexitätsstufen betrachtet werden. Dabei wird eine hierarchische Strukturierung geschaffen, die für das kognitive Problemlöseverhalten des Menschen geeignet ist. Aus der Psychologie ist bekannt, daß der Mensch Informationen und Aufgaben im Gedächtnis auf hierarchisch strukturierte Netzwerke abbildet, d.h. ein Problem in eine Summe von Teilproblemen zerlegt (Wippich 1984). Ein wichtiges Prinzip der ergonomischen Gestaltung ist die Abbildung der hierarchisch gegliederten Aufgabe auf die Benutzerschnittstelle. Dies bedeutet, daß die kognitive Struktur der Aufgabe als Modell für die Gestaltung der Mensch-Maschine Interaktion benutzt wird.
Dieses Vorgehen mit Hilfe der hierarchischen Strukturierung soll durch geeignete Baukastenobjekte in die Implementierung umgesetzt werden können. Es wird daher im Interfacebaukasten zwischen elementaren Bedienelementen (Klasse "Element") und Teilschnittstellen (Klasse "Constructor") unterschieden, die verschiedene Elemente zur Lösung einer Teilaufgabe zusammenfassen. Dies entspricht in der Hardwaretechnologie

einer Zusammenfassung von Bedienelementen (z.B. Schalter und Knöpfe) in einzelnen Systemmoduln (z.B. Einschübe).

Mit Hilfe dieser abstrakten Oberklassen (d.h. von diesen Klassen werden keine Instanzen gebildet) werden für jede Benutzerschnittstelle durch Vererbung Mechanismen bereitgestellt, die wesentlich zur Verwirklichung konsistenter Benutzerschnittstellen beitragen. Zusätzlich wird der Kodierungsaufwand der nicht-abstrakten Objekte reduziert ("code sharing"). Als gemeinsame Oberklasse dient die Klasse "Interactor" (Bild 3.3), in der Methoden und Instanzvariablen vereinbart sind, die beide Unterklassen gleichermaßen betreffen. Dies sind z.B. die Geometrie, die Hierarchiezugehörigkeit oder der Aktivierungszustand eines Objekts.

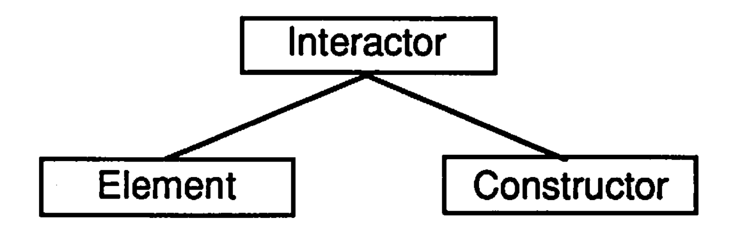


Bild 3.3: Oberklassen der Interaktionsobjekte.

3.2.1.1 Beschreibung der Oberklassen

Definition der Klasse "Constructor":
Konstruktoren fassen einzelne Elemente oder weitere Konstruktoren zusammen und organisieren deren Darstellung in einem eigenen Koordinatensystem. Sie dienen als Grundlage für die Definition komplexer Dialoge innerhalb einer Teilschnittstelle. Für die Benutzerschnittstelle entsteht dadurch eine Baumstruktur, deren Knoten von Konstruktoren (Teilschnittstellen) und deren Blätter von Elementen (Elementarschnittstellen) gebildet werden (Merkwitz 1989). Die abstrakte Baumstruktur einer hierarchisch aufgebauten Benutzerschnittstelle ist in Bild 3.4 dargestellt.

Innerhalb einer Benutzerschnittstelle sind Teilschnittstellen nicht zu jedem Zeitpunkt sichtbar bzw. zugreifbar; Konstruktoren können daher zwei verschiedene Aktivierungszustände annehmen:
1. "locked", d.h. die Teilschnittstelle steht nicht zur Interaktion bereit,
2. "standby", d.h. Interaktion ist innerhalb dieser Teilschnittstelle möglich.

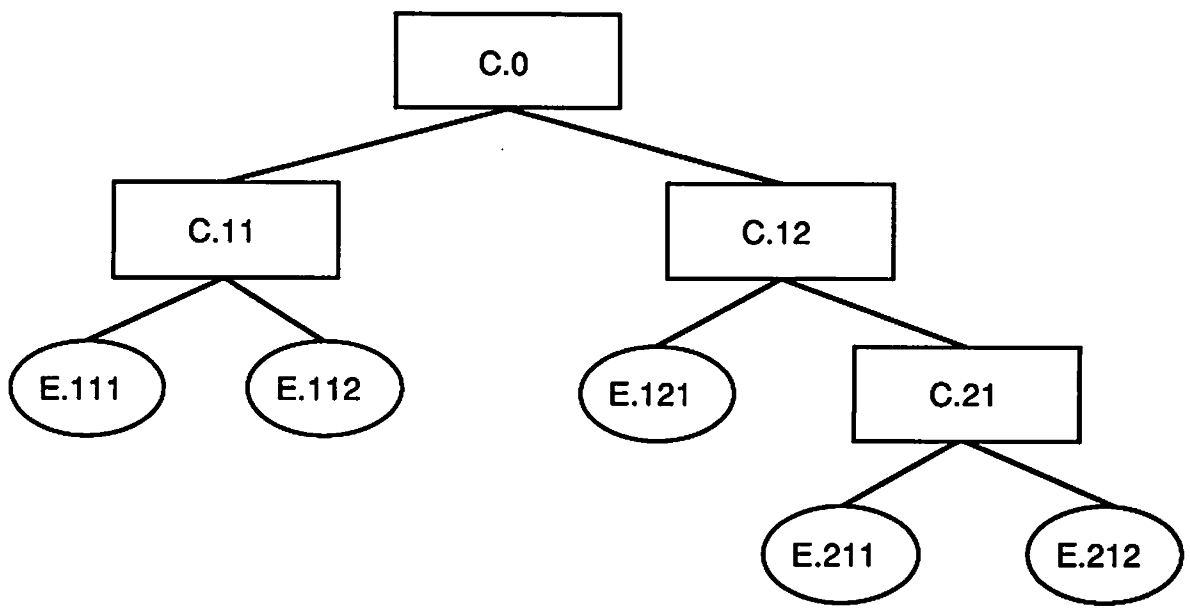


Bild 3.4: Hierarchische Baumstruktur einer Benutzerschnittstelle.

In Hinblick auf das evolutionäre Prototyping bieten Konstruktoren die Möglichkeit, zunächst einen speziellen Aufgabenbereich -also den Inhalt eines Konstruktors- zu bearbeiten und andere Teilaufgaben währenddessen durch leere Konstruktoren ("black boxes") zu vertreten (Top-Down Vorgehen). Ebensogut können zunächst verschiedene Teilaufgaben unabhängig voneinander bearbeitet werden und dann später zusammengefügt werden (Bottom-Up Vorgehen). Beide Methoden werden in der Systementwicklung eingesetzt, wobei die gewählte Methode von den Randbedingungen eines Projekts abhängt.

Definition der Klasse "Element":
Ein Element kann 3 verschiedene Aktivierungszustände annehmen, deren mögliche Zustandsübergänge (Transitionen) durch einen Zustandsautomaten beschrieben werden (Abschnitt 3.2.1.2). Es wird festgelegt, welche Interaktionsmodi (Maus-, Berühreingabe..) für ein Element jeweils zugelassen werden. Die Anbindung applikationsspezifischer Funktionen ist in jedem Element als "Slot" vorbereitet, so daß bestimmte Transitionen mit dem Aufruf dieser Funktionen gekoppelt werden können.

3.2.1.2 Herleitung des Zustandsautomaten

Die Interaktionsobjekte einer Schnittstelle sollen Eigenschaften aufweisen, die einer Anwendungswelt soweit wie möglich entsprechen. Im realen Arbeitsumfeld können bestimmte Objekte in beliebiger Reihenfolge bearbeitet werden, einige befinden sich bereits in der Bearbeitung und auf andere kann momentan nicht zugegriffen werden. Diese Sicht spiegelt sich in der Definition von drei charakteristischen Zuständen wider

(Klocke 1985), die nachfolgend mit "standby", "active" und "locked" bezeichnet werden. Diese Zustände können für den Benutzer durch Farbkodierung veranschaulicht werden, um so eine dynamische Benutzerführung zu ermöglichen (Rau und Trispel 1982).

- Zustand "locked" bedeutet, daß eine Interaktion mit dem Objekt nicht möglich ist, Eingaben werden ignoriert. Dieser Zustand wird benutzt, um bei der Durchführung einer Interaktionssequenz nur bestimmte Objekte zur Bedienung zuzulassen.
- Zustand "standby" eines Objekts sagt aus, daß dieses Objekt bereit ist, an der Interaktion teilzunehmen.
- Zustand "active" tritt ein, wenn ein Objekt durch eine Eingabe aktiviert wurde; das System erfährt dadurch eine vorübergehende oder bleibende Veränderung.

In den Interaktionselementen des Baukastens wird die Benutzerführung für alle Interaktionselemente in der jeweiligen Unterklasse genauer spezifiziert und implementiert. Für Tasten wird z.B. eine Farbkodierung des Hintergrunds vorgesehen, die die Zustände durch eine Helligkeits- und Sättigungskodierung visualisiert.

Zur einheitlichen Beschreibung der Zustände und der Steuerung der Übergänge kann ein endlicher, deterministischer Automat definiert werden. Bild 3.5 zeigt eine allgemeine Darstellung eines endlichen Automaten $A=<Q,I,O,t,m,q_0>$ mit Q als Zustandsmenge, I als Eingabealphabeth, O als Ausgabealphabeth, t die Transitionsabbildung, m die Ausgabeabbildung und q_0 der Ausgangszustand. Die Eingaben a_i des Automaten lösen eine Transition t_i aus, die in der Ausgabe O und einem neuen Zustand q' resultiert.

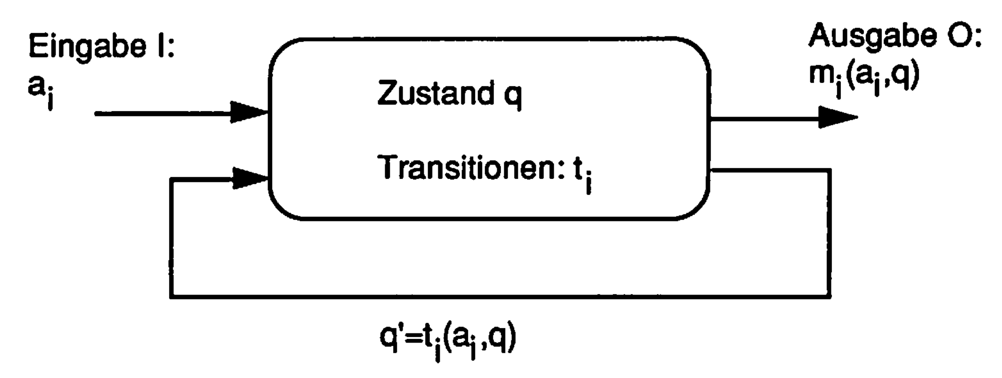


Bild 3.5: Endlicher Automat (Mealy-Automat).

Bild 3.6 zeigt nun die beschriebenen Zustände Q={locked, standby, active} und deren mögliche Übergänge t_i. Es handelt sich dabei um einen Mealy-Automaten, da die Ausgabe des Automaten nicht nur vom Ausgangszustand, sondern auch von der jeweiligen Transition abhängt.

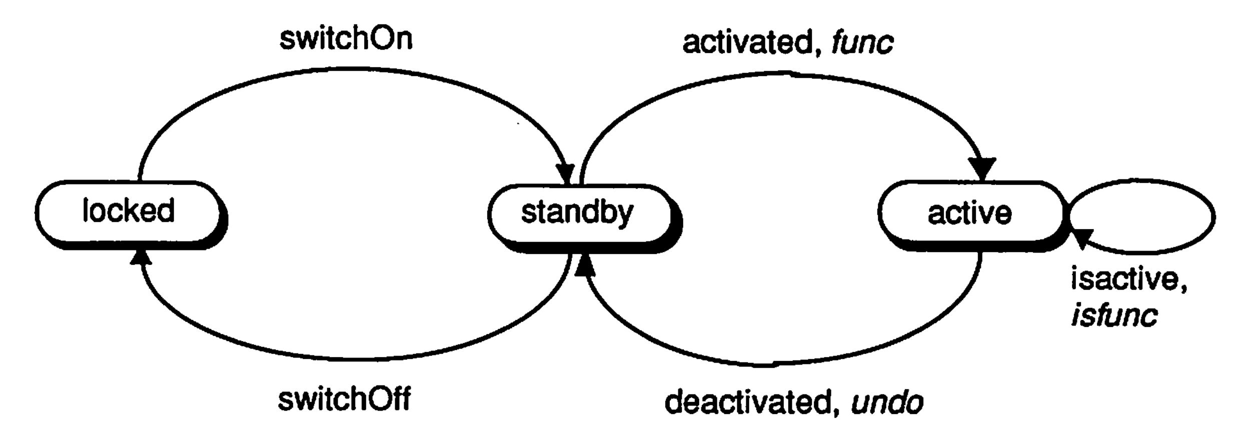


Bild 3.6: Zustände und Transitionen des Zustandsautomaten.

Die Transitionen "switchOn" und "switchOff" sind systemgesteuert und damit nicht unmittelbare Folge von Benutzerinteraktion. Sie gelten sowohl für "Element" als auch für "Constructor" und können aufgrund der Beschreibung des komplexen Dialogs von einem Dialogmanager vorgenommen werden.

Grundsätzlich hat jede Transition als Ausgabe ein visuelles Feedback. Die Transitionen "activated", "isactive und "deactivated" werden unmittelbar durch Aktionen des Benutzers ausgelöst und beschreiben damit den elementaren Dialog mit Interaktionselementen. Diese Transitionen können die angebundenen Funktionen *func*, *isfunc* bzw. *undo* als zusätzliche Ausgabe auslösen. Das Eingabealphabet wird für jedes Element in einer Liste festgegehalten, da ein Element nur an bestimmten Eingaben interessiert ist. Grundsätzlich kann ein Interaktionsobjekt jeden der vier Eingabemodi Maus, Tastatur, Berühr- und Spracheingabe benutzen. Das Eingabealphabet eines Werteschiebers im Zustand "active" bestehe z.B. aus dem Bewegen des Mauszeigers oder dem Loslassen der gedrückten Maustaste:

$$a_1 = \text{userInput(move)}, \; t_1 = \text{isactive};$$
$$a_2 = \text{userInput(release)}, \; t_2 = \text{deactivated}.$$

Eine applikationsspezifische Reaktion des Werteschiebers während des Bewegens wird in der Funktion *isfunc* definiert. Es kann an jedes Interaktionselement eine individuell angepaßte Funktion *func*, *isfunc* und/oder *undo* mitangebunden werden.

Durch den so definierten Automaten wurde erreicht, daß der elementare Dialog infolge von Benutzereingaben von jedem Interaktionselement intern verwaltet werden kann ohne daß zusätzliche externe Kontrollmechanismen geschaffen werden mußten.

3.2.1.3 Beschreibung der Unterklassen

Die in dieser Arbeit definierten und geschaffenen Unterklassen von "Element" und "Constructor" (Bild 3.7) werden im folgenden kurz vorgestellt.

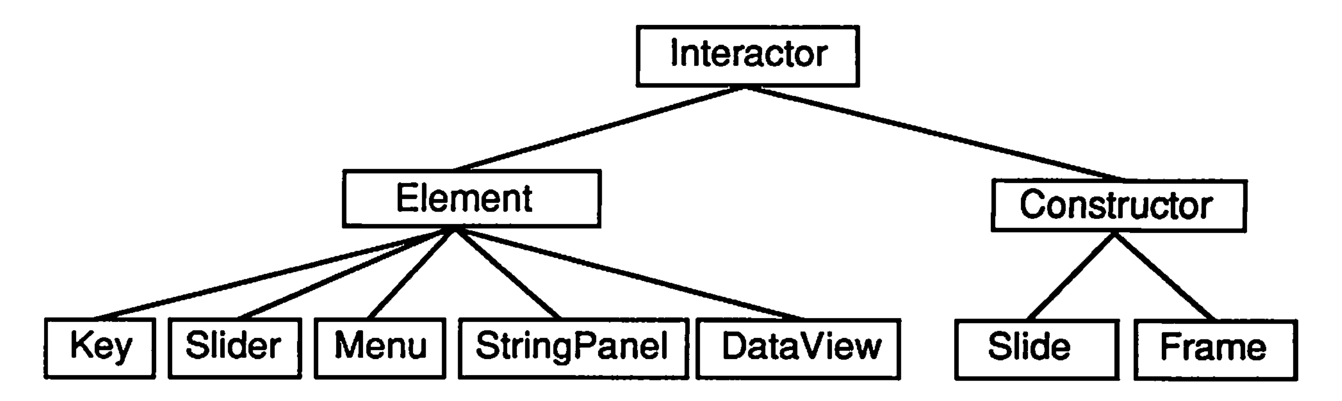


Bild 3.7: Objekthierarchie der Interaktionsobjekte.

Key: virtuelle Taste, die mit Text oder einem Piktogramm beschriftet werden kann.

Slider: analoger Werteschieber in horizontaler oder vertikaler Lage mit einstellbarer Skalierung.

Menu: Pulldown Menüleiste mit der Möglichkeit für angebundene Submenüs.

StringPanel: alphanumerische Eingabezeile, die durch Tastatur- oder Spracheingabe belegt werden kann.

DataView: xy- Diagramm zur Darstellung von Kurvenverläufen.

Frame: Teilschnittstelle, die permanent auch im nichtaktiven Zustand (locked) durch einen Rahmen angezeigt wird. Nur im aktiven Zustand (standby) werden die dazugehörigen Elemente dargestellt.

Slide: Teilschnittstelle, die - im Gegensatz zu "Frame" - wie ein Dia eingeblendet wird, d.h. im nichtaktiven Zustand unsichtbar ist.

Die Unterklassen von "Interactor" besitzen eine unterschiedliche graphische Repräsentation und benutzen daher intern unterschiedliche Grafikroutinen zur individuellen Darstellung; dennoch wird jedes Element durch Senden der Nachricht "display" zur Darstellung gebracht (Polymorphie). Die Darstellung einer Teilschnittstelle kann also einfach erreicht werden, indem der Konstruktor an alle ihm zugeordneten Objekte die Nachricht "display" sendet. Bei der Darstellung werden grafische Attribute (Farbe, Form, Beschriftung etc.) entsprechend vorheriger Vereinbarungen oder, falls dies nicht erfolgt ist, durch geeignete Voreinstellung ("sinnvoll gewählter Defaultwert") berücksichtigt. Im Abschnitt 3.3 werden die genannten Objekte innerhalb eines Beispiels benutzt, das einen genaueren Einblick in die Verwendung und Eigenschaften dieser Objekte gibt.

3.2.2 Präsentationsobjekte

In dieser Klasse werden Objekte sowohl für grafische als auch für akustische Ausgaben zusammengefaßt. Die Klasse der "PaintObjects" dient der grafischen Darstellung von Information, die nicht der direkten Manipulation zugänglich ist. Die Struktur der Hierarchie ist in Bild 3.8 dargestellt.

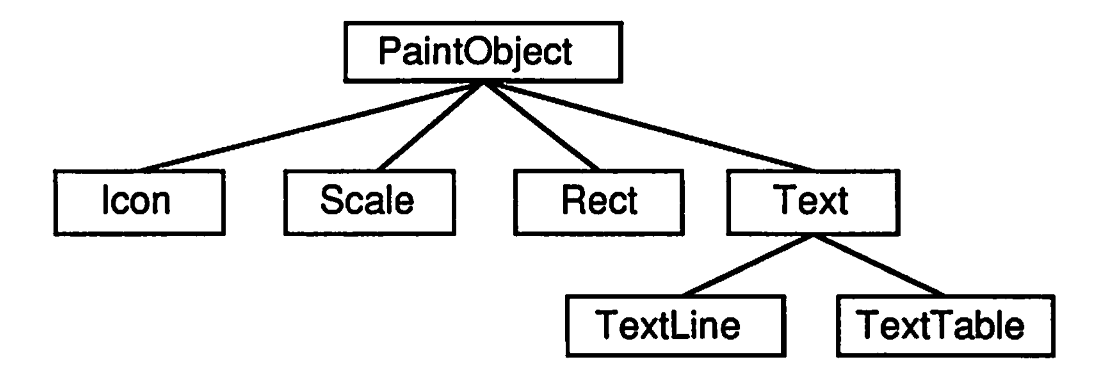


Bild 3.8: Objekthierarchie der "PaintObjects".

Jedes "PaintObject" wird - ebenso wie "Element" - einer Teilschnittstelle zugeordnet. Die Darstellung erfolgt abhängig vom Zustand der Teilschnittstelle ebenfalls polymorph durch die Nachricht "display", die für jedes Objekt individuell definiert ist. Die Präsentation von akustischer Information wird mit Hilfe des Objekts "Speaker" durchgeführt. Es ermöglicht die Verbindung mit einem Sprachausgabegerät, das digitalisierte Wörter und Wortketten oder synthetisch generierte Geräusche ausgibt. Die Erstellung von akustischen Signalen ist im Abschnitt 5.3 "Werkzeug zur Gestaltung akustischer Ausgabe" näher beschrieben.

3.2.3 Management multimodaler Interaktion

3.2.3.1 Modelle zur Beschreibung interaktiver Schnittstellen

Das Interaktionsmanagement einer Benutzerschnittstelle erfordert die Beschreibung durch ein zugrunde liegendes Modell. Derzeit werden vorwiegend drei Modelle verwendet:

- Beschreibung durch Zustandsübergangs-Diagramme,
- Beschreibung durch kontext-freie Grammatiken,
- Beschreibung durch das "*event*-Modell".

Zustandsübergangs-Diagramme beschreiben eine Schnittstelle als Verbund von Zuständen (Knoten), die durch Übergänge (Bögen) ineinander überführt werden

können. Die Schnittstelle befindet sich immer in genau einem definierten Zustand, der nur durch bestimmte Aktionen verlassen werden kann. Der Vorteil liegt in der leichten Verständlichkeit dieser Beschreibung; bei großen Systemen führt die Auflistung aller möglichen Zustände jedoch zu unübersichtlichen Darstellungen.

Kontext-freie Grammatiken sind eine regelorientierte Beschreibung für die möglichen Zustände und Aktionen an einer Schnittstelle; ein "Parser" kontrolliert die vorgegebene Grammatik. Die Stärke liegt in der Beschreibung linearer, textorientierter Sprachen, die Beschreibung von Dialogen in direkt-manipulativen Schnittstellen ist schwierig.

Das *event*-Modell betrachtet den Benutzer als Quelle von Ereignissen (*events*), die an unterschiedlichen Stellen der Benutzerschnittstelle über Eingabegeräte erzeugt werden können. Zu jeder Zeit können hier mehrere Zustände aktiv sein, die durch *events* verändert werden und dabei wieder neue *events* auslösen können. Das *event*-Modell ermöglicht eine für evolutionäres Prototyping direkt-manipulativer Schnittstellen effektive Beschreibungsform, die verständlich dargestellt und leicht in eine Implementierung umgesetzt werden kann. Die Darstellungsform der beiden anderen Modelle führt bei einer iterativen Vorgehensweise schnell zu unübersichtlichen Dokumenten (Farooq 1988). Für das *event*-Modell wurde in dieser Arbeit die Darstellung durch die Trennung in einen elementaren Dialog (Interaktionsmanagement) und einen komplexen Dialog (Dialogmanagement) erleichtert.

3.2.3.2 Client-Server Modell

Das "Client-Server" Modell ist eine spezifische Ausprägung des *event*-Modells. Es beschreibt ein System, in dem Anforderungen und Leistungen über ein Netzwerk von Rechnern verteilt sind. Innerhalb dieses Systems gibt es Erbringer bestimmter Leistungen (Server) und Nutznießer dieser Leistungen (Clients). Dieses Modell ist in dem Graphiksoftwaresystem X-Windows verwirklicht. Jede Graphikstation innerhalb eines Rechnernetzes wird von einem eigenständigen Prozeß, dem X-Server, verwaltet. Ein X-Server erbringt folgende Leistungen:

- Überwachung von Tastatur und Mauszeiger
- Verwaltung der Fenster und deren Attributen (properties)
- Verwaltung der Grafikressourcen in den Fenstern

Ein Anwendungsprogramm (Client) kommuniziert mit dem X-Server über einen Duplex-Byte-Stream, der innerhalb eines Netzwerks auf dem TCP/IP Protokoll basiert (Scheifler und Gettys 1986). Entgegen dem traditionellen Konzept "ein Bildschirm = ein Prozeß" kann der X-Server gleichzeitig mehrere Anwendungsprogramme bedienen (Bild 3.9).

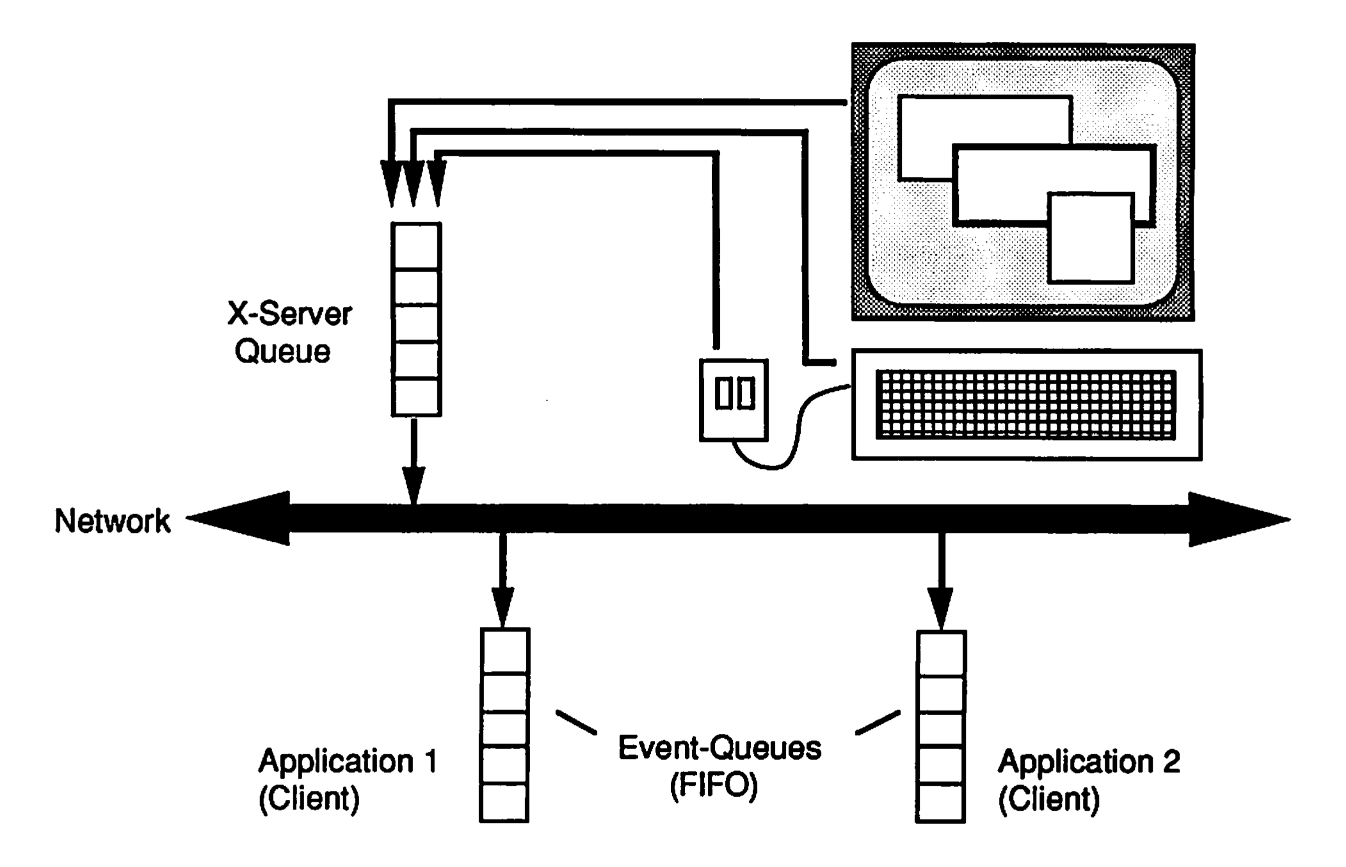


Bild 3.9: Client-Server Modell.

Unter einem *event* versteht man ein Ereignis, das von einer externen Quelle als Seiteneffekt eines Funktionsaufrufes asynchron erzeugt wird. Diese Ereignisse gelangen in die *event-queue* des X-Servers (First In First Out Speicher) und können dann in der Reihenfolge ihres Auftretens bearbeitet werden. Die Datenstrukturen der *events* verfügen über gleiche Teilstrukturen, wie z.B. die *event*-Typkennung, an der die *events* unterschieden werden können. Anhand des *event*-Typs kann eine Fallunterscheidung aufgebaut werden, in der dann der *event*-spezifische Teil der Datenstruktur ausgewertet wird.

Das Kernstück interaktiver Programmsysteme bildet das Abfragen und reagieren auf Benutzereingaben (sog. Ablaufsteuerung). Diese Funktionalität fehlerfrei zu implementieren erfordert vom Applikationsprogrammierer komplexer Systeme einen großen mentalen und zeitlichen Aufwand. Schon bei der Auswertung von für das Fenstersystem X-Window spezifischen *events* (z.B. Mausklick) entstehen umfangreiche "if" oder "case" Verschachtelungen, die erfahrungsgemäß leicht zu Fehlerquellen des Programmsystems werden.

3.2.3.3 Struktur der Server im Interfacebaukasten

Im Konzept des Interfacebaukastens sollen neben den für X-Window typischen Eingabekanäle Maus und Tastatur zusätzlich weitere Eingabekanäle eingebunden werden. Die Berücksichtigung dieser Kanäle durch Abfragen in einer Schleife (Polling) widerspricht dem Konzept des *event*-Modells und die Verarbeitung der Ereignisse wäre nicht synchronisiert. Dieser gravierende Nachteil läßt sich nur vermeiden, wenn das Konzept einer Integration aller Eingabekanäle verwirklicht wird und dadurch alle Ereignisse in einem Kanal gebündelt werden. Dieses Konzept sollte so angelegt werden, daß der Applikationsprogrammierer vollständig (!) von der *event*-Verwaltung innerhalb der Ablaufsteuerung entlastet wird.

Die folgenden Anforderungen wurden an das Konzept eines Interaktionsmanagers gestellt:

- Eingabeverwaltung nach dem *event*-Modell in einem eigenständigen Handler-Objekt,
- Konzentration aller *events* in einer Ereignisschlange (Synchronisation),
- Aufrechterhaltung der Netzwerkfähigkeit,
- Adressierung von Nachrichten an die Empfängerobjekte,
- Steuerung der *event*-Verwaltung konsistent zu den X-Window Standardmechanismen (wie z.B "Focus" und "Grab").

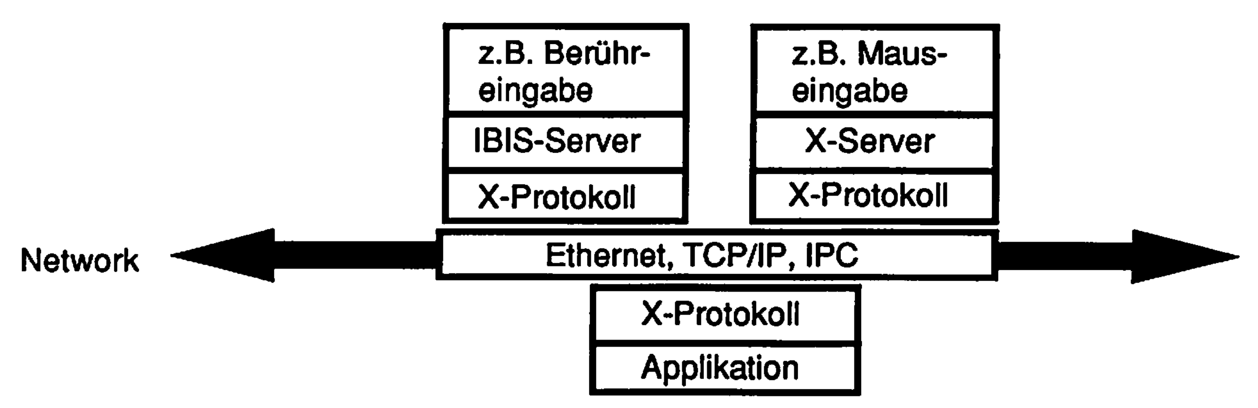


Bild 3.10: Architektur des Integrationskonzeptes.

Im Lösungskonzept ist die Integration zusätzlicher Eingabekanäle (Multimodalität) in das X-Window Konzept durch zusätzliche Server verwirklicht worden (Schröter 1989). Sogenannte "IBIS-Server" überwachen je einen Eingabekanal und wandeln die dort erhaltenen Daten in ein Format um, das zu der X-Eventstruktur kompatibel ist (X-ClientMessages). Durch diese Abbildung auf eine X-kompatible Datenstruktur wird dann automatisch die Forderung nach Netzwerkfähigkeit und Synchronisation in einer X-*event-queue* erfüllt (Bild 3.10).

Die Überwachung und Auswertung der *event-queue* übernimmt der "EventController" (Bild 3.11). Dieses Objekt des Interfacebaukastens ist als Zentrum des Interaktionsmanagements anzusehen, da es die Verbindung zwischen den Ereignissen an einem Eingabegerät und den betroffenen Objekten einer Applikation herstellt. Jedem *event* wird ein Empfänger zugeordnet, der die Nachricht einer Eingabe in Form sog. "input token" erhält. Eine Applikation trägt hierfür alle Interaktionsobjekte einer Benutzerschnittstelle in eine Liste des "EventControllers" ein, in der jeweils das Objekt und für das Objekt selektierte Eventtypen zugeordnet sind. Der "EventController" bestimmt mit Hilfe dieser Liste für jedes *event* das Empfängerobjekt und adressiert ein "input token" direkt dorthin. Die weitere Verarbeitung dieses *events* obliegt dem Empfängerobjekt. D.h. der elementare Dialog wird vollständig nach dem *event*-Modell bearbeitet; die auf den Zustandsautomaten bezogenen Übergänge ergeben sich dann aus dem Zusammenspiel der im EventController und in den Objekten verkapselten (impliziten) Informationen.

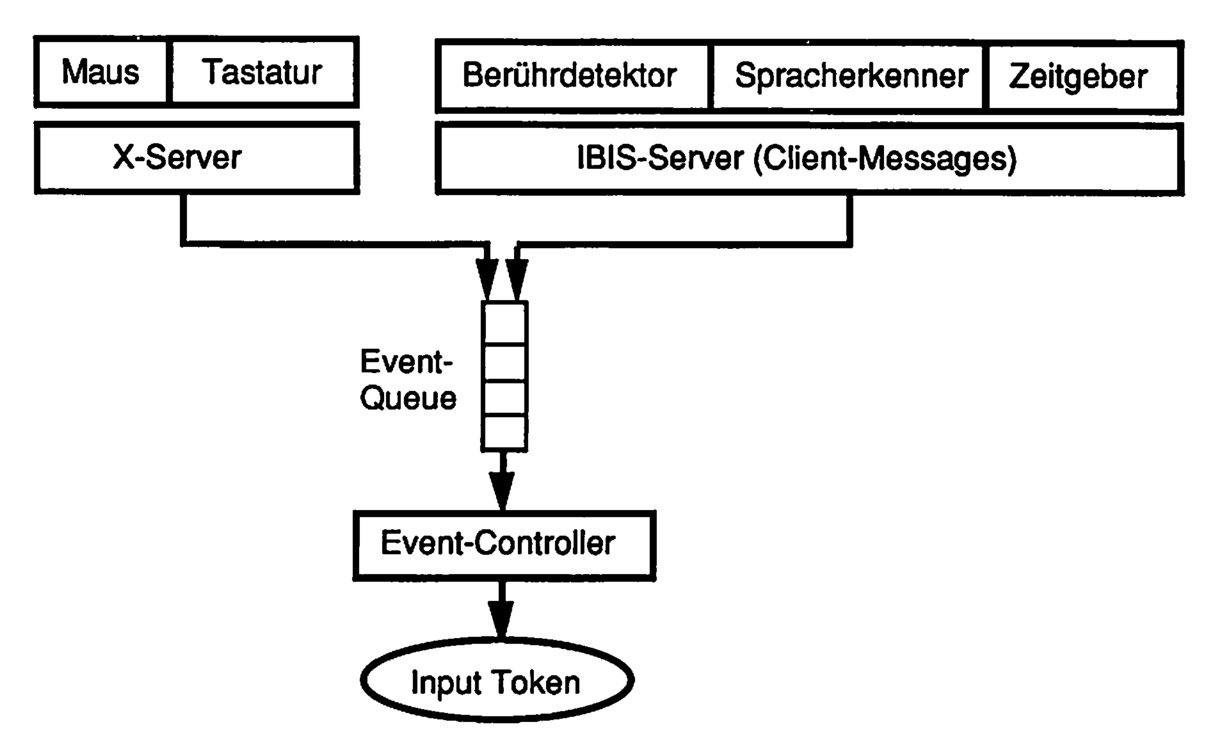


Bild 3.11: Konzentration von Interaktionsereignissen.

Die Objekt-Struktur der "IBIS-Server" wurde hierarchisch aufgebaut, um physikalische und logische Geräte zu trennen. Dadurch wird eine Erweiterung oder der Austausch von Geräten (z.B. anderes Spracherkennungssystem) leicht möglich. Objekte der logischen Geräteebene behandeln die kanalspezifischen Eigenschaften (z.B. das Versenden von Koordinatenpaaren für Berühreingaben). Der gerätespezifische Anteil wird unabhängig davon in einem Objekt der physikalischen Geräteebene behandelt ("Gerätetreiber"). Folgende Server wurden implementiert (Bild 3.12):

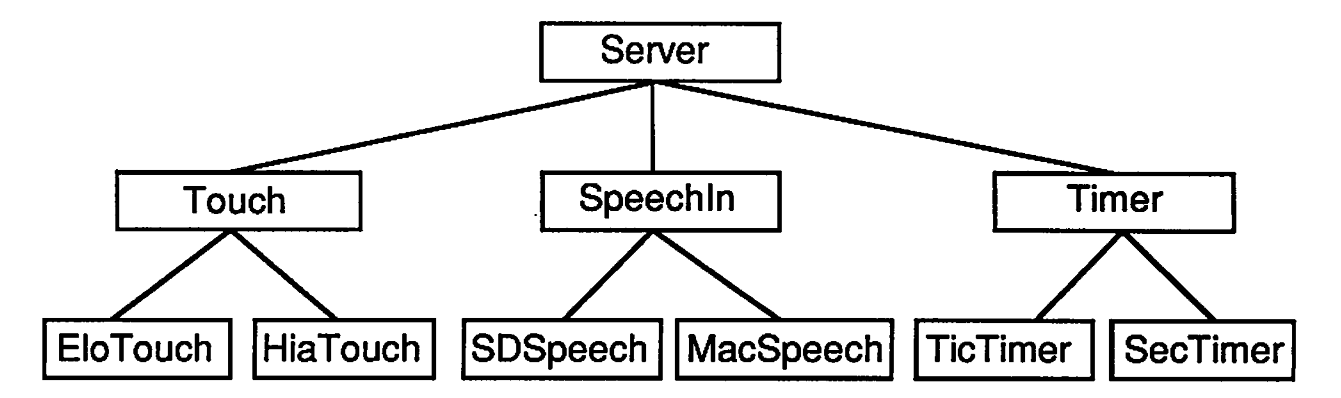


Bild 3.12: Hierarchie der IBIS-Server.

- Berühreingabe: Der Server versendet ein oder (bei 2-Finger Berührung) zwei Koordinatenpaare, die bereits auf ein Standard-Koordinatensystem umgerechnet sind. Es kann ein Foliendetektor (EloTouch) oder ein Infrarot-Lichtschrankendetektor (HiaTouch) angeschlossen werden.
- Spracherkennung: Der Server versendet die erkannten Wörter als ASCII-String. Zwei verschiedene Spracherkennungssysteme (Macintosch Voice Navigator™ oder Speech-Design™) können angeschlossen werden.
- Zeitgeber: Dieser Server sendet in einem definierten Zeittakt Nachrichtenimpulse, mit denen frei definierbare Funktionen getriggert werden können. Die Zeitauflösung kann in Sekunden (SecTimer) oder Millisekunden (TicTimer) gewählt werden. Diese Möglichkeit macht das Gesamtkonzept nach außen hin offen und damit besonders flexibel. Die Ankopplung externer Prozesse oder auch die Animation einer Schnittstelle können durch den Zeitgeber realisiert werden.

Das multimodale Interaktionsmanagement ist aus der Sicht des Applikationsprogrammierers im Objekt "EventController" vollständig verkapselt. Die Nachricht "startLoop" aktiviert die interne Laufzeitschleife, die dann die Gesamtheit aller Interaktionen in jeder Teilschnittstelle automatisch verwaltet, und somit die komplexe Ablaufsteuerung vollständig realisiert. Der Programmkode erhält dadurch eine Struktur, die einfacher und schneller implementiert und leichter gewartet werden kann.

3.3 Anwendungsbeispiel: Entwurf und Implementierung einer Benutzerschnittstelle für eine Infusionspumpen-Kontrolleinheit

Die Gestaltung einer Benutzerschnittstelle für eine konkrete Aufgabe wird nun zur Überprüfung der Leistungsfähigkeit des Konzeptes des Interfacebaukastens IBIS vorgestellt. Am Beispiel der Gestaltung einer "Kontrolleinheit" für eine Infusionspumpe soll gezeigt werden, wie methodisch und durch Verwendung der Bausteine eine von der Aufgabenanalyse über die Spezifikation bis hin zur Implementierung durchgängig aufgabenbezogene Sichtweise die Entwicklung einer neuen Benutzerschnittstelle vereinfacht. Eine wesentliche Unterstützung bietet dabei die im Interfacebaukasten festgelegte Implementierungssprache DIC (Deskriptiver Interface Code).

3.3.1 Aufgabenanalyse

Gemeinsam mit dem Benutzer wird zunächst aus der Aufgabenanalyse heraus eine schematische Spezifikation angefertigt. Aus Benutzersicht ist dies eine Beschreibung seiner Aufgabe, aus Sicht des Systementwicklers ein erster Ansatz für die Struktur der Implementation.

Eine hierarchische Strukturierung führt zur übersichtlichen Darstellung von Teilsystemen, deren Komponenten von Benutzer und Entwickler auf jeder gewünschten Ebene diskutiert werden können. Die hierarchische Zerlegung wird zwischen zwei aufeinander folgenden Ebenen durch einen Baum dargestellt und kann sich sowohl auf Objekte als auch auf Aktionen beziehen. Da die Zahl der hierarchischen Ebenen variabel ist, können auch Systeme hoher Komplexität durch eine angemessene Anzahl von Ebenen detailliert betrachtet werden. In Bild 3.13 sind die Teilaufgaben festgelegt, die bei der Überwachung und Steuerung einer Infusionspumpe zu erfüllen sind.

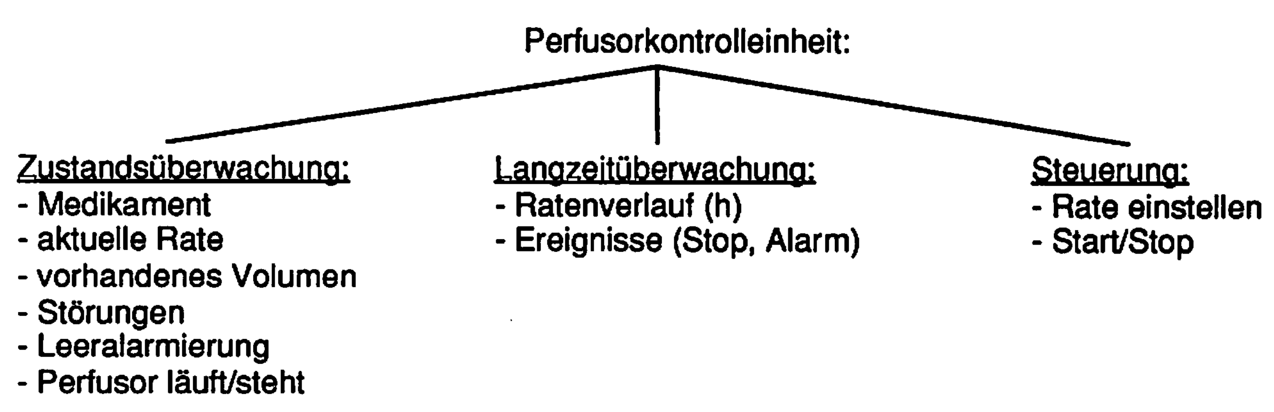


Bild 3.13: Aufgabenstruktur für die Kontrolle einer Infusionspumpe (Perfusor).

Mit Hilfe des Interfacebaukastens ist es nun möglich, aus der Aufgabenbeschreibung sukzessive das erste prototypische Labormuster abzuleiten. Dabei wird im Gegensatz zu Phasenmodellen ein fließender Übergang von Spezifikation, Entwurf und Implementierung verwirklicht. Die Implementierung erfolgt in 5 Schritten, die jeweils eine bestimmte unterschiedliche Sichtweise auf die Mensch-Maschine Schnittstelle wiedergeben (Bild 3.14).

I:	Auswahl von Interaktionsobjekten und Festlegung der geometrischen Struktur
II:	Anpassung der Objektmerkmale (Darstellungs- und Interaktionsmodi)
III:	Vervollständigung der Darstellung durch Präsentationsobjekte (Beschriftungen, Piktogramme etc.)
IV:	Definition des komplexen Dialogs (Dialogmanagement)
V:	Einbinden der Datenmanagement-Funktionen (Applikationsspezifische Funktionalität)

Bild 3.14: Vorgehensweise bei der Implementierung.

Wenn diese Schritte im Quellkode nacheinander jeweils für alle Objekte einer Schnittstelle implementiert werden, ist dies ein Top-Down Entwurf. Es können jedoch auch jeweils für jedes Objekt alle Schritte zusammengefaßt werden und zunehmend mehr Objekte derart aneinander gereiht werden (Bottom-Up Entwurf). Nachfolgend werden diese verschiedenen Schritte anhand des Beispiels erläutert.

3.3.2 Abfolge der Implementierung

Zur Bearbeitung der in Bild 3.13 veranschaulichten drei Teilaufgaben müssen zunächst Interaktionselemente ausgewählt und in einer geeigneten Geometrie angeordnet werden. Folgende Elemente werden gewählt:

Zustandsüberwachung: *alphanumerische Eingabe/Anzeige* von aktuellem Medikament und eingestellter Rate ("StringPanel", "Text")

Anzeige des vorhandenen Volumens als *Wertechieber* ("Slider")

Anzeigen von Störungen und Alarmen durch *Piktogramme* ("Icon")

Langzeitüberwachung: *Kurvendiagramm* mit Markierungen für Ereignisse
("DataView").
Elemente der Steuerung:
Start/Stop durch *Taste* ("Key")
Rate steuern:
Dosis-Anwahl durch *Taste* ("Key")
dieser Taste untergeordnet werden:
Dosis-Einstellung durch *Werteschieber* und Enter-*Taste*
("Slider", "Key").

3.3.2.1 Schritt I

Die aufgabenbezogene Vorgehensweise kann durch Einsatz der Baukastenobjekte in einer objektorientierten Implementierung beibehalten werden. Der (aufgabenunbezogene) Umgang mit Graphikroutinen und Event-Abfragen, die sonst in jedem Applikationsprogramm notwendig sind, sind in der Objekt-Welt des Baukastens verkapselt. Die Implementierung wird mit Hilfe einer objektorientierten Kodierung verwirklicht, die nicht nur compilierbaren Quellkode, sondern gleichzeitig auch eine anschauliche Beschreibung der Benutzerschnittstelle darstellt. Dieser "Descriptive Interface Code" (DIC) verbindet so die Leistungsfähigkeit sog. Pseudo-Codes hinsichtlich der Lesbarkeit mit der Übersetzbarkeit einer Sprache wie C. DIC baut auf der Objective-C Syntax auf und folgt dem Schema:

```
[Empfängerobjekt   Nachricht_1: Argumente_1 ...   Nachricht_n: Argumente_n]
```

Als Beispiel wird die Implementierung der Taste beschrieben, die die Teilschnittstelle zum Einstellen der Infusionsrate aktiviert. Die Programmzeile:

```
doseKey = [Key   create: geometry   childOf: InfusionControl]
```

sagt aus, daß eine Taste mit der gegebenen Geometrie ("geometry" ist eine Rechteckstruktur) innerhalb (als "child") der Teilschnittstelle "InfusionControl" angezeigt werden soll. "InfusionControl" ist ein Objekt der Klasse "Constructor" und verwaltet die Teilschnittstelle der Steuerungselemente. Allein diese Programmzeile reicht aus, um Darstellung und elementare Interaktionsverwaltung dieser Taste zu berücksichtigen. Durch Setzen der charakteristischen Hintergrundfarbe für den Zustand "aktivierbar" wird automatisch die Benutzerführung realisiert. Ebenfalls durch Voreinstellung (default) ist diese Taste sensitiv für Berühr- und Mauseingabe, d.h. diese Taste reagiert auf diese *events* mit der Transition in den Zustand "aktiviert" und ändert dann automatisch die Farbkodierung zur Benutzerführung.

Für ein erstes Layout, in dem nur die geometrische Strukturierung der Benutzerschnittstelle untersucht werden soll, ist damit bereits eine ausreichende Kodierung erreicht. In (Bild 3.15) ist die geometrische Anordnung aller Interaktionsobjekte zu sehen. Die insgesamt horizontale Ausrichtung ist optisch einer realen Spritzenpumpe angenähert.

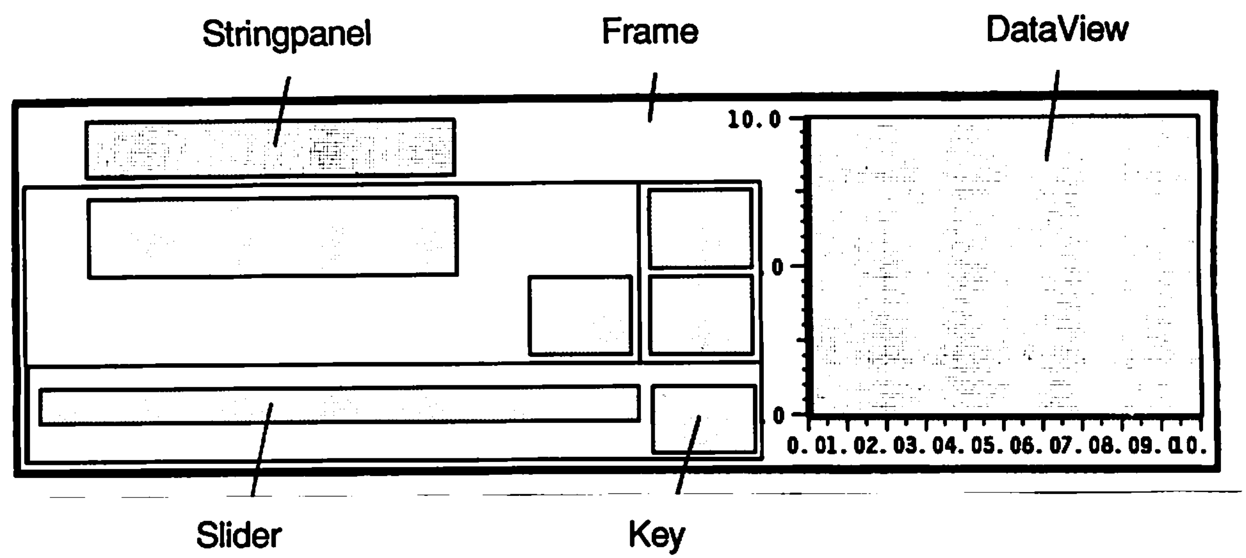


Bild 3.15: Schnittstelle nach Schritt I.

3.3.2.2 Schritt II

Zur weiteren Charakterisierung der Schnittstelle können nun durch Ausführung von Methoden bestimmte Attribute von Interaktionsobjekten gesetzt werden. Im nächsten Schritt soll z.B. eine Beschriftung der Taste vorgenommen werden.

```
[doseKey setLabel:"dose.pic"]
```

Die Nachricht "setLabel" mit dem Argument "dose.pic" bewirkt, daß die Datei dose.pic als Quelldatei einer Bitmapgrafik gelesen wird und das darin definierte Piktogramm zentriert in der Taste erscheint. Die Zeile

```
[doseKey setLabel:"Dosis"]
```

hingegen hätte den genannten Text "Dosis" als Beschriftung eingesetzt. Diese Alternativen zeigen, daß dem Anwendungsprogrammierer die Betrachtung der völlig verschiedenen Grafikkontexte für die Darstellung von Piktogrammen und Texten erspart wird. Es wird seine Sichtweise berücksichtigt, indem das Setzen *einer* Beschriftung mit Hilfe des "Descriptive Interface Code" (DIC) sowohl für ein Piktogramm als auch für Text sehr ähnlich ist.

Die Berücksichtigung der Spracheingabe als zusätzlichen Interaktionsmodus bewirkt die Anweisung:

[doseKey setBEvents:"SpeechIn: Rate"];

das akustische Kommando "Rate" überführt die Taste in den Zustand "active". Die einzige Voraussetzung hierfür ist, daß dieses Wort innerhalb eines Wortschatzes trainiert wurde, der dem Spracheingabe-Server bekannt gegeben wurde. Die Vor- und Hintergrundfarben sind ebenfalls Attribute, die in Schritt II gesetzt werden. Nach Schritt II hat die Benutzerschnittstelle das in Bild 3.16 dargestellte Aussehen.

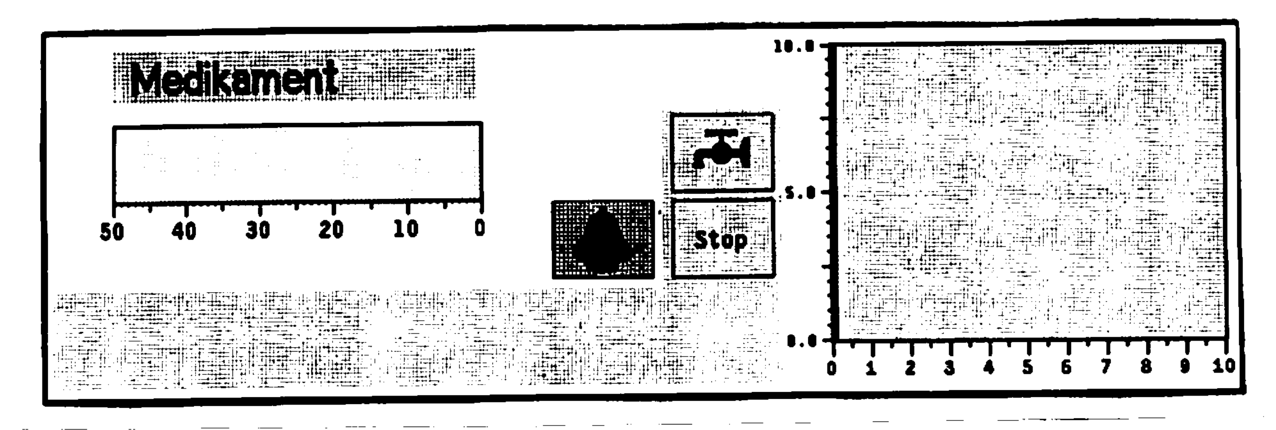


Bild 3.16: Schnittstelle nach Schritt II.

3.3.2.3 Schritt III

In Schritt III wird das Aussehen der Schnittstelle vervollständigt, indem zusätzliche Beschriftungen und Piktogramme hinzugefügt werden. Beispielweise wird der Infusionsschlauch durch ein in der Farbe der Infusionslösung dargestelltes Rechteck repräsentiert. Die Anweisung dazu lautet.

```
InfusionTube = [Rect  create: geometry   colorOf: Injector
                      partOf: InfusionDisplay]
```

Der Infusionsschlauch gehört zur Teilschnittstelle "InfusionDisplay". Seine Hintergrundfarbe ist mit der Farbe der Spritze "Injector" identisch. Als Ergebnis von Schritt III ensteht Bild 3.17. Damit sind Präsentation und Interaktionsmanagement der Benutzerschnittstelle vollständig beschrieben. Es fehlt jetzt noch die Definition der Dialogkomponente und die Anbindung an den Prozeß durch applikationsspezifische Funktionen.

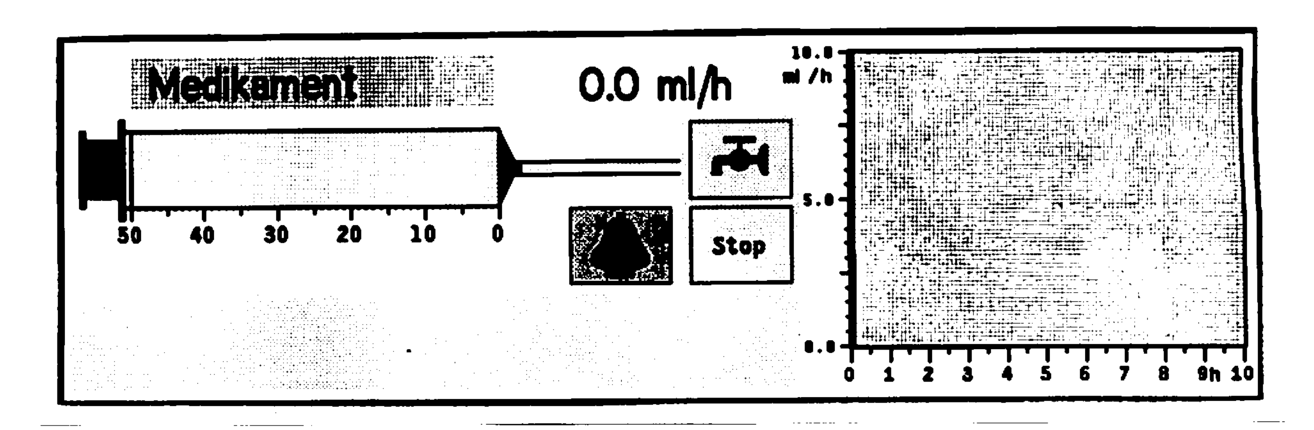


Bild 3.17: Schnittstelle nach Schritt III.

3.3.2.4 Schritt IV

Die Dialogkomponente einer Benutzerschnittstelle definiert Interaktionssequenzen, die zum Zwecke der Benutzerführung die Zustände von Objekten beeinflußen. Im Beispiel der Infusionspumpen-Kontrolleinheit erscheint die Teilschnittstelle "adjustment" mit Werteschieber und Enter-Taste nur, wenn vorher die Dosis-Taste aktiviert wurde. Dieser Zusammenhang wird dem Dialogmanager durch folgende Eingabe bekannt gemacht:

```
[InfusionControl  setLinkFrom: doseKey   To: adjustment
                              on: STANDBY with: "SwitchOn"]
```

D.h., wird die Dosis-Taste im Zustand "standby" aktiviert, so wird die Teilschnittstelle "adjustment" eingeblendet ("SwitchOn"). Farbbild 3.1 (Anhang B-1) zeigt die Schnittstelle nach der Aktivierung der Dosis-Taste; in diesem Zustand sind der Werteschieber und die "Enter"-Taste aktivierbar.

3.3.2.5 Schritt V

Die Transition vom Zustand "standby" in den Zustand "active" kann eine applikationsabhängige Funktion auslösen, wenn diese Funktion vorher durch eine entsprechende Anweisung an ein Objekt angebunden wurde. Durch die Anweisung

```
[enterKey  setFunc: F_enterKey]
```

wird die freidefinierbare C-Funktion "F_enterKey" an die Taste "enterKey" angebunden, der Taste zur Bestätigung der eingestellten Dosis. So kann erreicht werden,

daß bei der Aktivierung dieser Taste z.B. der eingestellte Wert zur Protokollierung in eine Datenbasis übertragen wird.

Ebenso wird in Schritt V der Datenfluß von der Applikation zur Benutzerschnittstelle festgelegt. Farbbild 3.1 zeigt, wie eine Störung des Perfusors (offene Zuleitung) an der Benutzerschnittstelle durch ein Piktogramm visualisiert wird.

Damit konnte am Beispiel der Gestaltung einer "Kontrolleinheit" für eine Infusionspumpe gezeigt werden, wie durch methodisches Vorgehen in fünf Schritten und durch Verwendung der Bausteine die Entwicklung einer neuen, multimodal bedienbaren Benutzerschnittstelle vorgenommen werden kann. Dabei konnte ausgehend von der Aufgabenanalyse mit wenigen Zeilen Programmkode eine aufgabenbezogene Implementierung durchgeführt werden.

3.4 Beurteilung des Interfacebaukastens

Die in Abschnitt 3.1.1 gesetzten Ziele für den Interfacebaukasten wurden in der vorliegenden Implementierung erreicht:

- Die Möglichkeit zur **hierarchischen Strukturierung** einer Benutzerschnittstelle wurde durch die Einführung von "Konstruktoren" erreicht.
- Die **Konsistenz einer Benutzerschnittstelle** wird durch die vorgefertigten Bausteine unterstützt, in denen feste Mechanismen verkapselt sind. Dadurch wird die Flexibilität zwar eingeschränkt, gleichzeitig wird aber auch die Vielfalt möglicher Fehler stark verringert, so daß der Entwurf ergonomisch hochwertiger Schnittstellen erleichtert wird.
- Die **multimodale Interaktion** wurde durch die Integration zusätzlicher *event-Server* für die Eingabekanäle Spracherkennung und Berühreingabe erreicht. Auch die akustische Ausgabe kann über einen entsprechenden Baustein auf einfache Weise eingebunden werden.
- Eine **aufgabenbezogene Quellkodierung** wird durch die neu eingeführte Semantik des DIC (deskriptiven Interfacekode) erreicht.

Der Interfacebaukasten ist ebenso wie ein Applikationsprogramm ein Software-Produkt, das mit Hilfe der Kriterien des Software-Engineering bewertet werden kann (siehe Abschnitt 1.3). Es werden daher diese Kriterien und zusätzlich die von Löwgren (1988) formulierten Anforderungen an ein "user interface management system" (UIMS) in die nachfolgende Bewertung miteinbezogen.

3.4.1 Benutzbarkeit

Die Frage, wie einfach ein UIMS benutzt werden kann, wird anhand der erforderlichen Programmierkenntnisse beantwortet, die zur Erstellung einer Applikation notwendig sind. Im Extremfall sind keine Programmierkenntnisse nötig wie z.B. im System Peridot (Myers 1987), wo der Benutzer eine Schnittstelle mit Hilfe von Beispielen entwirft. Dies bedeutet aber eine Einschränkung bei der Entwicklung komplexer Schnittstellen. In solchen Fällen ist ein UIMS verbunden mit geringen Programmieranforderungen leistungsfähiger. Im Sinne einer flexiblen Entwicklungsumgebung sollte nicht auf eine strukturierte, textuelle (z.B. objektorientierte) Repräsentation verzichtet werden (Perlman 1988). Die textuelle Repräsentation der Benutzerschnittstelle wird durch die in Abschnitt 3.3 beschriebene Implementierungssprache DIC verwirklicht. Die Vorteile einer graphischen Programmierung können mit dem im Abschnitt 4.2 beschriebenen Entwurfseditor genutzt werden. Die Kombination dieser beiden Vorgehensweisen stellt einen guten Kompromiß für hohe Flexibilität und geringe Programmieranforderungen dar.

3.4.2 Allgemeinheit

Dieses Kriterium soll zunächst hinsichtlich der Interaktionsformen angewendet werden, die von einem UIMS unterstützt werden. Fast alle bekannten UIMS zeigen in diesem Punkt starke Einschränkungen, weil lediglich Tastatur und Maus im Mittelpunkt stehen. Nach Myers (1987b) sollten jedoch folgende Punkte erfüllt werden:

- Unterstützung verschiedener interaktiver Techniken,
- Erweiterungsmöglichkeit für die Integration neuer Interaktionstechniken,
- verschiedene Interaktionstechniken zur selben Zeit verfügbar,
- verschiedene Interaktionstechniken zur selben Zeit operierend,
- geeignete einfache Sprache zur Programmierung bzw. grafische Programmierung.

Diese Punkte sind durch das Konzept des multimodalen Interaktionsmanagement (Abschnitt 3.2.3) berücksichtigt worden.

Eine Beschränkung der bekannten UIMSs besteht darin, daß der Entwurf einer Benutzerschnittstelle für nur einen Bildschirm ermöglicht wird. Das vorgestellte Baukastensystem IBIS hingegen unterstützt auch den Entwurf von Mehrbildschirmsystemen innerhalb eines Netzwerkes.

3.4.3 Portabilität

Die Portabilität kann durch Verwendung von Standards in der Hardware und Software erreicht werden. Der Interfacebaukasten benutzt folgende (z.T. de facto) Standards:
- Betriebssystem UNIX
- Programmiersprache C unter UNIX
- Graphiksystem X Windows Version 11.3

Die Erweiterungen zur akustischen Ein-/Ausgabe sind hardwareabhängig. Keine der objektorientierten Sprachen hat sich bislang als Standard durchgesetzt. Die im Interfacebaukasten verwendete Sprache "Objektive-C" bietet die Möglichkeit, Quellkode in die Sprache C zu übersetzen; dieser C-Kode ist dann wieder portabel.

3.4.4 Wartbarkeit und Erweiterbarkeit

Für den Systemdesigner stellt das UIMS eine "Black-Box" dar, in die keine Eingriffe möglich sind. Diese Einschränkung ist sinnvoll, da der Systemdesigner nicht notwendigerweise mit Details der Implementierung vertraut sein muß.
Die Wartung und Erweiterung ist Aufgabe der Entwickler des UIMS. Jedes Objekt sollte einem Betreuer zugeteilt werden, der alleine berechtigt ist, Änderungen vorzunehmen. Der zusätzliche Aufwand der Wartung eines UIMS wird durch die Einsparung an Wartungsaufwand für die damit erstellten Programme mehr als ausgeglichen; in konventioneller Programmierung erfordert jedes erstellte Applikationsprogramm einen programmiererfahrenen Betreuer.

3.4.5 Verläßlichkeit

Die Verläßlichkeit eines Objekts steigt mit der Anzahl der erfolgreichen Einsätze eines Objekts in unterschiedlichem Kontext. Die Eigenschaft der Wiederverwendbarkeit erlaubt es, die "Reife" eines Objekts durch diese Anzahl anzugeben. Jede Änderung eines Objekts vermindert diese Reife wieder. Wenn aber nur ein erfahrener Betreuer zu Änderungen berechtigt ist, wird auch bei einer Änderung die Verläßlichkeit eines Objekts nur geringfügig geringer.

3.4.6 Effizienz

Laufzeiteffizienz:
Die objektorientierte Programmierung erfordert einen zusätzlichen Verwaltungsaufwand. Das Senden einer Nachricht (=Auslösen einer objektorientierten Methode) benötigt etwa die doppelte Zeit gegenüber einem konventionell programmierten Funktionsaufruf. Die Verwendung von Objekten des Interfacebaukastens führt auf einer Workstation jedoch zu keiner sichtbaren Verzögerung der Eingabe-Reaktionen (Feedback). Zum Starten einer Anwendung benötigt die Initialisierung der Objekte ca. 20-40 Sekunden[1] (je nach Größe der Applikation). Die Starttotzeiten in der genannten Größenordnung treten auch in anderen bausteinorientierten Programmierumgebungen (z.B. HP-Widgets) auf.

Speicherplatzeffizienz:
Es ist gegenüber konventioneller Programmierung eine Steigerung des Platzbedarfs aufgrund der Anbindung einer flexiblen Bibliothek zu beobachten. Der Speicherbedarf der Infusionspumpen-Kontrolleinheit beträgt etwa 280 kByte. Diese Größe ist bei einer heute üblichen Arbeitsspeicherkapazität von einigen MByte und Plattenkapazitäten von einigen hundert MByte unbedeutend.

3.4.7 Wirtschaftlichkeit

Die Kosten der Softwareentwicklung im Prototyping-Verfahren steigen zunächst an, da in die Entwicklungsumgebung zunächst einmal Zeit bzw. Geld investiert werden muß. Insgesamt ergibt sich jedoch auf lange Sicht durch wiederholtes Anwenden der Entwicklungsumgebung eine gesteigerte Wirtschaftlichkeit, insbesondere durch die Einsparungen bei den heute noch immens hohen Wartungskosten (Boehm 1984).

3.4.8 Produktivität

Eine Zielsetzung von UIMSs ist die Produktivitätssteigerung in der Softwareentwicklung. Darunter versteht man die Verbesserung des Verhältnisses Produktumfang zu Entwicklungskosten sowie der Qualität. Die Produktivität wird durch die Wiederverwendung von Software gesteigert; eine Formel zur Berechnung der Produktivität unter Berücksichtigung von Häufigkeit und Grad der Wiederverwendung beschreibt Endres (1988).

[1] UNIX Workstation HP 9000-340

Eine eigene Studie zur Produktivitätssteigerung mit Hilfe des Interfacebaukastens wird anhand folgender Softwareentwicklungen dargelegt:

1. Das Anwendungsbeispiel der Infusionspumpen-Kontrolleinheit konnte mit 170 Zeilen implementiert werden. Eine nichtoptimierte Implementierung ohne Baukastenobjekte konnte abgeschätzt werden, indem die benutzten Methoden der verwendeten Objekte wie eine implementierte C-Funktion angerechnet wurden. Nach dieser Schätzung wären für diese Schnittstelle 850 Zeilen konventioneller Kode notwendig gewesen; die Einsparung liegt also bei 80 Prozent.
2. Der in Abschnitt 4.4 beschriebene Farbeditor wurde in der ersten Entwicklung mit C und X-Windows implementiert und im Verlauf einiger Monate erweitert und optimiert. Der Quellkode umfaßt ohne die Farbraumtransformation 2200 Zeilen. Die gleiche Funktionalität wurde später mit Hilfe des Interfacebaukastens implementiert und umfaßt 1500 Zeilen. Die Kodereduktion gegenüber der bereits optimierten Version betrug somit 32 Prozent.

Derartige Schätzungen der Produktivität anhand der Programmzeilen sind vielfach kritisiert worden, weil wichtige Kriterien wie die Komplexität eines Programms, die Art der Programmierung (Programmiersprache) und die Qualität nicht geeignet berücksichtigt werden. Die Produktivitätsformel nach Case (1985) versucht auch diese Kriterien zu berücksichtigen, indem auch Komplexität der Software, Anzahl der nach einer Installation gefundenen Fehler und die zur Korrektur benötigte Zeit betrachtet werden. Auch dieser Aspekt der Produktivität wird durch den Interfacebaukasten positiv beeinflußt, weil

- die installierte Software aufgrund der Reife wiederverwendbarer Bausteine weniger Fehler enthält,
- die Zeit zum Auffinden und Korrigieren von Fehlern aufgrund einer stark verbesserten Programmlesbarkeit kürzer ist und
- die Kodekomplexität verringert wird, weil die Anzahl der Zeilen und die Schleifenverschachtelung geringer ist.

4. Werkzeuge zur Informationsdarstellung

Die Informationsdarstellung am Bildschirm umfaßt verschiedene Möglichkeiten der Kodierung, z.B. durch Farbe, Form, Text oder Piktogramme. Der effiziente Einsatz dieser Kodierungsmöglichkeiten erfordert Kenntnisse über die Informationsverarbeitung des Menschen, um die perzeptiven und kognitiven Fähigkeiten und Grenzen durch eine geeignete Gestaltung der Informationsdarstellung und Interaktion berücksichtigen zu können.

In diesem Kapitel werden daher zunächst Aspekte der perzeptiven und kognitiven Fähigkeiten des Menschen näher betrachtet, die dann in den Konzepten der Werkzeuge für Bildschirmentwurf, Farbgestaltung und Piktogrammgestaltung berücksichtigt werden.

4.1 Menschliche Informationsverarbeitung graphischer Darstellungen

4.1.1 Modelle menschlicher Informationsverarbeitung

Es existieren verschiedene Modelle der menschlichen Informationsverarbeitung, die die perzeptiven und kognitiven Prozesse näher beschreiben. Card, Moran und Newell (1983) entwickelten ein Modell, das in Anlehnung an die Rechnertechnologie interpretiert werden kann (Bild 4.1).

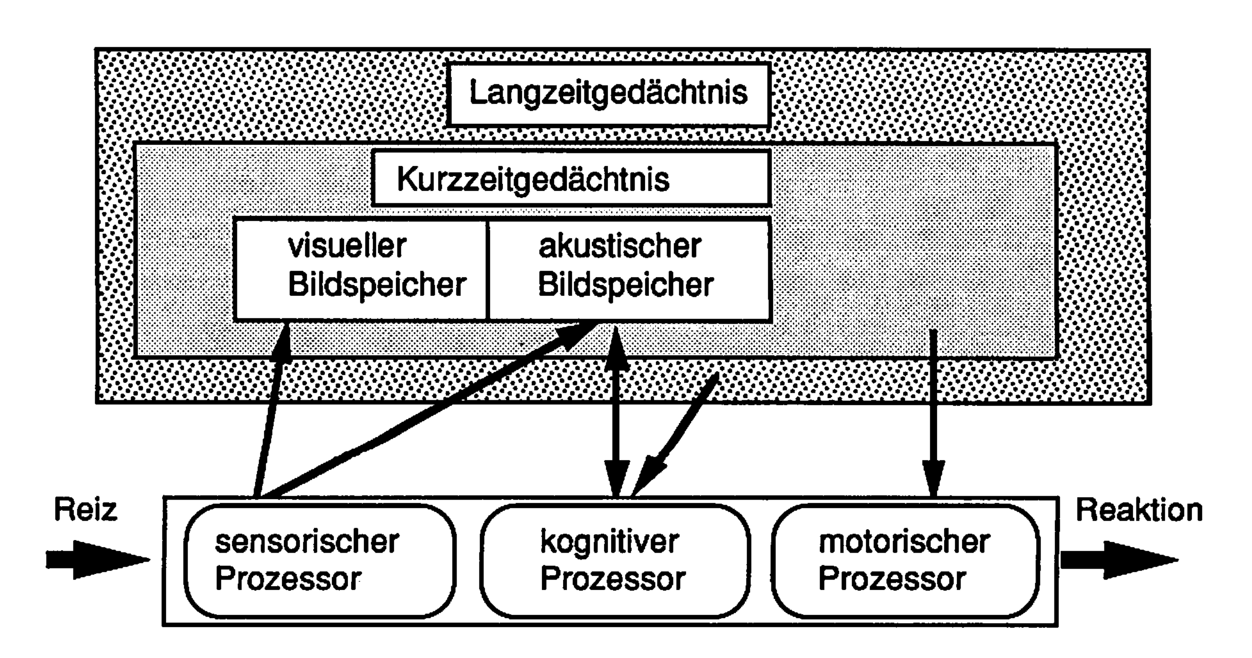


Bild 4.1: Modell der Informationsverarbeitung nach Card et al. (1983).

In diesem Modell menschlicher Informationsverarbeitung werden drei Prozessoren unterschieden, der sensorische, motorische und kognitive Prozessor; es werden für jeden Prozessor Speicherkapazität, Zeitkonstante der Speichererhaltung, Zugriffszeit und Zykluszeit einer Verarbeitung definiert und ihre Größe in Versuchen ermittelt. Im Modell wird nur eine serielle Informationsverarbeitung berücksichtigt: Information gelangt zunächst über den sensorischen Prozessor in das Kurzzeitgedächtnis (Arbeitsspeicher); die Weiterverarbeitung übernimmt der kognitive Prozessor, wobei dieser auf das Kurzzeitgedächtnis und Langzeitgedächtnis zugreift. Soll eine Reaktion erfolgen, so wird diese durch den kognitiven Prozessor eingeleitet und dann vom motorischen Prozessor ausgeführt. Die Bearbeitungszeit wird im Mittel mit 100 ms angegeben und bezieht sich auf einfache Prozesse (z.B. das Bedienen eines Taschenrechners). Bei komplexen Vorgängen reicht diese Bearbeitungszeit des kognitiven Prozessors nicht aus; die "Komplexität" des Vorgangs kann dann als multiplikative Größe berücksichtigt werden (Hick 1952).

Das von Card et al. entwickelte Modell basiert auf der Vorstellung einer Ein-Kanal-Verarbeitung, die nur bedingt zutreffend ist. Ein Modell, das auch eine parallele Verarbeitung berücksichtigt, wurde von Wickens (1984) entwickelt (Bild 4.2).

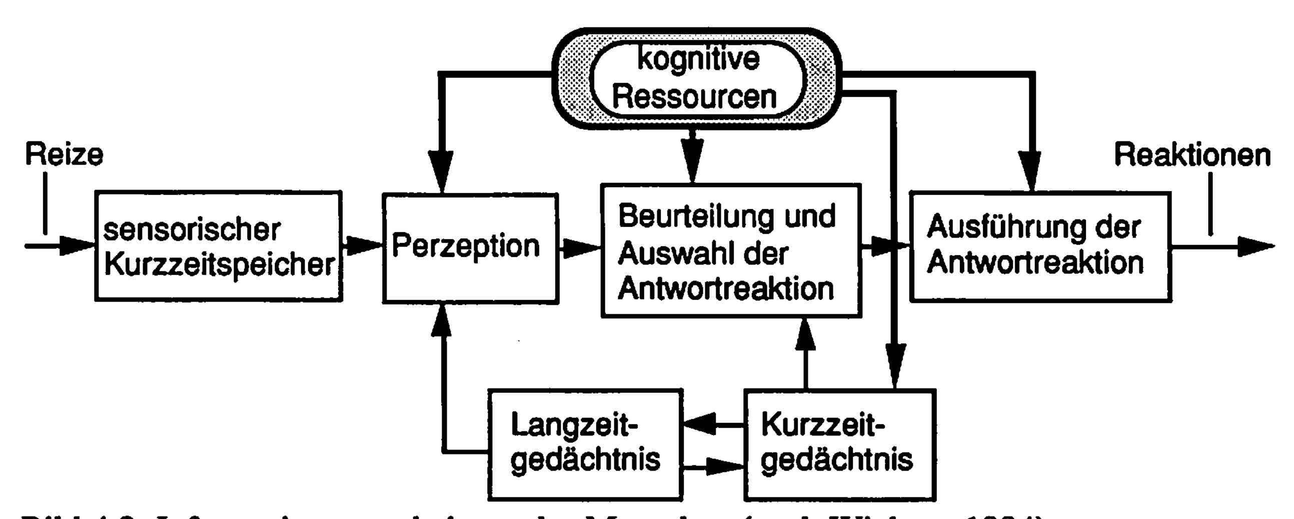


Bild 4.2: Informationsverarbeitung des Menschen (nach Wickens 1984).

Auch hier werden vom Reiz bis zur Reizantwort drei Schritte durchlaufen:
1. Reizaufnahme/Reizerkennung (Stimulus-Perzeption),
2. Reizanalyse und Auswahl der Antwortreaktion,
3. Ausführung einer Reaktion.

Während dieser Verarbeitung wird auf das Gedächtnis und die sogenannten "kognitiven Ressourcen" zugegriffen, die als Basis der Leistungsfähigkeit des menschlichen Gehirnes bei der Informationsverarbeitung angesehen werden. Diese kognitiven

Ressourcen können parallel verschiedenen Prozessen zugeteilt, jedoch nicht für jeden Prozeß gleich effektiv eingesetzt werden. In der "Stimulus - Central Processing - Response" Theorie wird erläutert, daß bestimmte Parameter eine "natürliche" Repräsentation besitzen und dadurch mit der kognitiven Verarbeitung kompatibel sind (Moses et al 1979). So wird die Gefährlichkeit eines Stoffes durch Farbe und die Menge durch die Größe eines Symbols geeignet repräsentiert. Wenn genau umgekehrt kodiert wird, so steigen Antwortzeit und Fehlerrate deutlich an. Ebenso muß die räumliche Kompatibilität einer Darstellung beachtet werden, die die Zusammengehörigkeit von Anzeigen, Eingabeelementen und Sensoren bestimmt.

Neben dieser "Reiz-Reaktion Kompatibilität" wird die Verarbeitung von Information durch Beschränkungen in der Aufmerksamkeit bzw. Aufnahme von Signalreizen (Perzeption) und Beschränkungen in der Analyse der aufgenommenen Signale (z.B. mangelnde Gedächtniskapazität) beeinflußt. Die Beschränkung der menschlichen Informationsverarbeitung wurde von Miller (1956) mit dem Gesetz "The Magical Number Seven Plus or Minus Two" beschrieben. Es besagt, daß innerhalb eines sensorischen Kontinuums nur eine diskrete Anzahl von 7±2 Werten sicher unterschieden werden kann.[1] Diese Ergebnisse sind auch auf nichtgleichwahrscheinliche Ereignisse und kognitive Prozesse übertragbar. Dazu wird der mittlere Informationsgehalt eines Signals - abhängig von der Anzahl der verschiedenen Ereignisse und deren Wahrscheinlichkeiten - nach folgender Formel berechnet:

$$H_g = \Sigma\, p(x_i)\, \mathrm{ld}\, (1/p(x_i))\ \mathrm{bit} \qquad (4.1)$$

Der Informationsgehalt eines Signals, das 10 verschiedene gleichwahrscheinliche Erscheinungsformen hat (z.B. ein Symbol mit einer von 10 verschiedenen Formen), berechnet sich als $H_g = 3.3$ bit.
Kann der Empfänger ein Signal nicht sicher identifizieren, so ist der übermittelte Informationsgehalt H_t um den Betrag des Informationsverlustes H_v geringer.

$$H_t = H_g - H_v \qquad (4.2)$$

Den Zusammenhang zwischen H_t und H_g gibt Bild 4.3 wieder. Er bezieht sich auf die sichere Erkennbarkeit eines isoliert präsentierten Reizes aus einer bekannten Menge (d.h. in einem Experiment werden mehr als 95 % der Reize richtig erkannt). Bei einer unbegrenzten Kapazität des menschlichen informationsverarbeitenden Kanals würde

[1] Diese Zahl ist vom Reiz abhängig. So können bei der Bewertung der Salzigkeit eines Nahrungsmittels nur 4 Abstufungen sicher unterschieden werden, während für die Bestimmung der Lage eines Punktes auf einer Geraden 11 Unterscheidungen getroffen werden können.

kein Informationsverlust auftreten (gestrichelte Linie). In der Realität wird jedoch -je nach Art des stimulierenden Reizes- die sichere Erkennbarkeit eines Signals im Bereich von 2 bis 3 bit (4 bis 8 Ereignisse) kritisch.

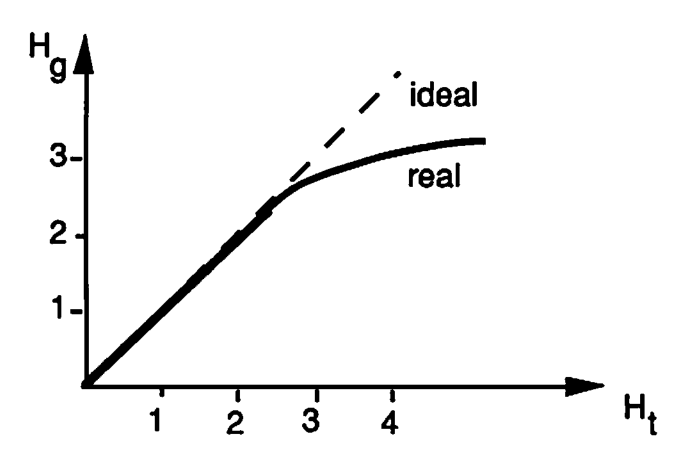


Bild 4.3: Sichere Erkennbarkeit einer Anzahl von Ereignissen aus Wickens (1984).

4.1.2 Visuelle Wahrnehmung

Räumliche Information wird in der rechten Hälfte des Gehirns verarbeitet; die Aufnahme erfolgt überwiegend holistisch, d.h. Dinge werden als Ganzes wahrgenommen. Der visuelle Kanal kann mehrere Reize parallel verarbeiten, sofern sie räumlich genügend nahe zusammenliegen. Die Organisation visueller Information steht im Mittelpunkt der Frage, wie die vom sensorischen System detektierte Information zur Wahrnehmung unserer Umgebung führt (Pomeranz 1986). Die Aufnahme eines visuellen Reizes wird insbesondere durch dessen Kontur (Kanten) beeinflußt. Korrelierte Attribute (Redundanz) unterstützen die Reizverarbeitung ebenso wie der Grad der Vertrautheit eines Symbols (Kompatibilität).

Ein visueller Reiz hat verschiedene visuelle Variablen (Bertin 1982):
- Ort eines Elements,
- Größe,
- Form,
- Orientierung,
- Farbton,
- Helligkeit,
- Muster, Textur und Rasterung.

Diese Variablen können nicht voneinander unabhängig wahrgenommen werden, z. B. hängt die Wirkung von Farbton oder Form von der Größe und Helligkeit ab.

Die mehrdimensionale Kodierung benutzt verschiedene visuelle Variable innerhalb eines Signals (z.B. Farbe und Form). Dabei unterscheidet man die orthogonale Kodierung, bei der die Dimensionen unabhängig voneinander interpretiert werden, von der korrelierten Kodierung, wo ein eindeutiger Zusammenhang zwischen den Dimensionen besteht. Der Informationsgehalt eines Signals mit einer aus 10 möglichen Formen und 10 verschiedenen Farben wird bei orthogonaler Kodierung auf 2*3.3 = 6.6 bit vergrößert. Wenn die beiden Dimensionen korreliert benutzt werden, so bleibt der Informationsgehalt von 3.3 bit bestehen. Da die Kanalkapazität von 6.6 bit nicht vollständig ausgenutzt wird, besteht eine Redundanz. Diese Redundanz verbessert nach Eriksen und Hake (1955) die Informationsübermittlung, man spricht daher von einem Redundanzgewinn. Durch korrelierte Kodierung läßt sich also die Sicherheit der Informationsübermittlung auf Kosten der Effizienz (Kanalausnutzung) steigern. Ebenso steigert die redundante Kodierung die Geschwindigkeit der Informationsverarbeitung, wie z. B. durch kombinierte Farb- und Formkodierung (Kopala 1979).

Eine integrierte Anzeige faßt verschiedene Kodierungsdimensionen in einer Darstellung zusammen (z.B. Farbe und Größe eines Rechtecks). Bei redundanter Kodierung wird ein Redundanzgewinn erzielt, bei orthogonaler Kodierung hingegen ist mit einem Informationsverlust gegenüber einer getrennten Anzeige zu rechnen, da genaues Ablesen von Werten auf getrennten Anzeigen leichter möglich ist (Wickens 1984). Integrierte Anzeigen mit orthogonaler Kodierung sind geeignet, wenn lediglich ein grober Überblick (qualitativ, ohne exakte Werte) über mehrere Parameter erwünscht ist. Auch für die medizinische Diagnose können diese integrierten Anzeigen von Vorteil sein, wenn dadurch die Korrelation verschiedener Werte besser sichtbar wird.

Beispiele für integrierte, orthogonal kodierte Anzeigen sind das "face display" (Bild 4.4a), in dem verschiedene Werte in Größe und Form von Merkmalen des Gesichts verschlüsselt werden (Jacob et al. 1976), sowie das "polar display" (Bild 4.4b), das für die Parameterüberwachung eines Reaktor-Kühlsystems entworfen wurde (Woods et al. 1981). Im ersten Beispiel wird die (täglich trainierte) Fähigkeit des Menschen ausgenutzt, die Mimik seiner Mitmenschen interpretieren zu können, bzw. bekannten Menschen bestimmte Gesichter zuordnen zu können. Im "polar display" werden Fähigkeiten der Mustererkennung, bzw. die Empfindlichkeit der visuellen Wahrnehmung für Asymmetrien ausgenutzt.

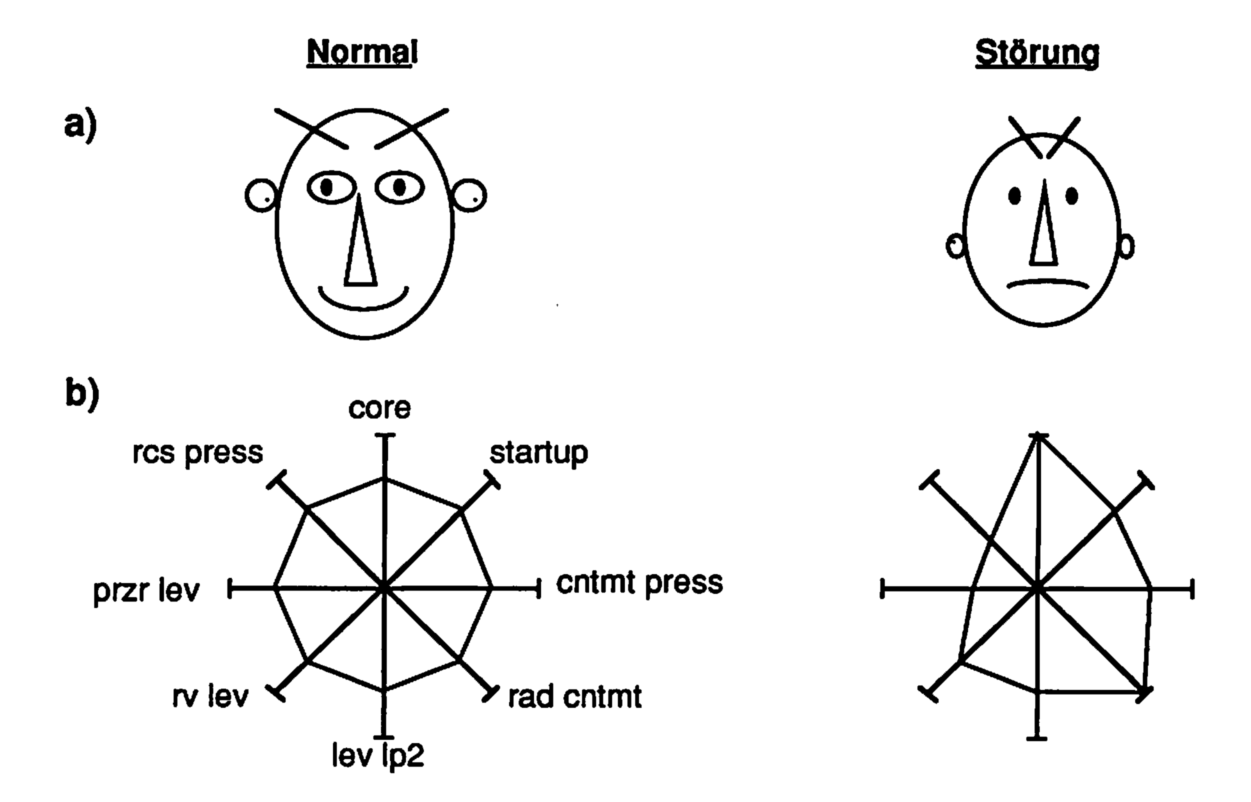


Bild 4.4: a) "Face Display". b) "Polar Display".

4.1.3 Gestaltgesetze

Das visuelle System interpretiert eine Darstellung unbewußt nach bestimmten Wahrnehmungsregeln, durch die die sensorische Information kontextabhängig weiterverarbeitet wird. In Bild 4.5 werden 3 gefüllte Kreise und 2 überlagerte Dreiecke wahrgenommen, die nur virtuell vorhanden sind, d.h. dargestellt sind nur 3 Kreissegmente und 3 Winkelelemente.

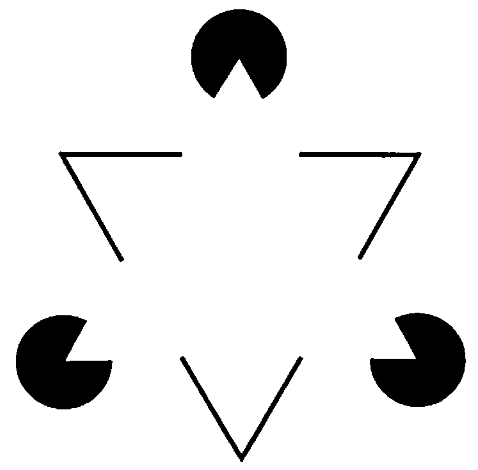


Bild 4.5: "Kanizsa-Figur".

Dieses Phänomen wird durch die Gesetze der Geschlossenheit und Fortführung aus der *Gestalttheorie* beschrieben (Wertheimer 1923). Diese Theorie definiert eine Gestalt als eine Gesamtheit, die sich von der Summe der Einzelheiten unterscheidet. Anstelle einer einfachen Superposition tritt also eine nichtlineare Interaktion von einzelnen Elementen.

Gesetz der Prägnanz:
Im Gehirn wird die sensorische Information so organisiert, daß ein möglichst einfaches Muster abgespeichert werden kann. Ähnlich einer Seifenblase, die automatisch die stabilste Form als Kugel annimmt, wird bei verschiedenen Interpretationsmöglichkeiten die jeweils einfachste ausgewählt und "gesehen".

Gesetz der Fläche:
Bei Überlagerungen wird angenommen, daß ein kleineres Objekt über einem größeren liegt.

Gesetz der Nähe:
Elemente, die nahe zusammenliegen, werden als Gruppen gesehen.

Gesetz der Gleichartigkeit:
Elemente, die z.B. in Farbe oder Form ähnlich sind, werden als Gruppe gesehen.

Gesetz der Geschlossenheit:
Das perzeptive System füllt nicht geschlossene Darstellungen selbst aus, da geschlossene Formen prägnanter sind als offene Formen.

Gesetz der Fortführung:
Ein unterbrochener Linienverlauf wird als kontinuierliche Linie mit Überlagerungen interpretiert.

Gesetz konvexer Formen:
Konvexe Formen werden eher zusammenhängend (als Figur) gesehen als konkave Formen.

Gesetz der Symmetrie:
Symmetrische Flächen werden als Figur bzw. Gruppe wahrgenommen. Symmetrie ist eine hervorspringende Eigenschaft und gilt als Schlüssel für die Güte eines Musters.

Weitere Prinzipien der Bildgestaltung lassen sich aus Kriterien der Werbung in Printmedien ableiten (Nelson 1981):

1. Ein Bild soll hinsichtlich der visuellen Variablen (Größe, Schwarzheit, Buntheit) ausbalanziert wirken.
2. Die Unterteilung des Bildes soll in angenehme Proportionen vorgenommen werden, die in Beziehung zueinander stehen.
3. Ein Bild soll wie durch einen Rahmen als Einheit zusammengehalten werden.
4. Eine Hervorhebung soll im Bild dominieren.

Zusammenfassend besitzt eine"gute Gestalt" eine einfache Struktur, deren Komponenten (a) in der Anzahl gering, (b) eine regelmäßige Form haben und (c) Symmetrien besitzen.

Damit wurden in diesem Abschnitt verschiedene charakteristische Merkmale des Menschen als informationsverarbeitendes System näher beleuchtet. Das Wahrnehmungsverhalten des Menschen insbesondere in bezug auf die visuelle Informationsverarbeitung wird nun in den folgenden Abschnitten in den Konzepten zum interaktiven Schnittstellenentwurf, zur Farbgestaltung und zur Piktogrammgestaltung berücksichtigt. Die graphische Informationsdarstellung soll dabei so organisiert werden, daß die Leistungsstärke des Menschen bei der Mustererkennung bestmöglich eingesetzt werden kann.

4.2 Werkzeug zum interaktiven Schnittstellenentwurf

Bereits der erste Entwurf einer neuen Benutzerschnittstelle ist im Detail betrachtet ein iterativer Prozeß; auch wenn die Art und Anzahl von Schnittstellen-Objekten feststeht, sind vielfältige Variationen in Größe und Ort eines Objekts möglich. Die im letzten Abschnitt genannten qualitativen Prinzipien bezüglich der Gestaltung lassen sich nicht "im ersten Wurf" realisieren; erfahrungsgemäß müssen die Objekt-Koordinaten häufig geändert werden, bis eine geeignete Anordnung gefunden ist. Jede Änderung erfordert dabei bisher eine erneute Übersetzung und Ausführung des Applikations-Programmes. Diese Vorgehensweise ist einerseits zeitaufwendig und andererseits ist der unmittelbare Vergleich zweier Versionen nicht möglich.

Eine deutlich Verbesserung kann erreicht werden, wenn ein interaktiver Entwurf in der Weise möglich wird, daß interaktive Schnittstellen ähnlich wie CAD-Graphiken am Bildschirm erstellt werden können. An einen derartigen interaktiven Entwurfseditor werden nun folgende Anforderungen gestellt, die aus der Erfahrung beim Schnittstellenentwurf mit Hilfe des Interfacebaukastens entstanden sind:

- Interaktives Erzeugen von Interaktionsobjekten aus dem Interfacebaukasten; Positionieren und Festlegung der Größe mittels variierbarer temporärer Umrandungen ("rubber banding").
- Interaktive Modifikation der Objektgeometrien.
- Simulation der Benutzerinteraktion während des Entwurfs.
- Generierung von Quellkode (DIC), der von dem aktuellen graphischen Entwurf zu einem lauffähigen Programm führt.
- Entwurf von Mehrbildschirm-Arbeitsplätzen. Das Werkzeug soll auf einem seperaten Bildschirm bedient werden können.
- Quantitative Analyse und Bewertung der geometrischen Anordnung von Objekten auf dem Bildschirm.
- Nachträgliche online-Modifikation laufender Applikationsprogramme.

Die Realisierung dieser Anforderungen in dem interaktiven Entwurfseditor der Software-Entwicklungsumgebung (vgl. Bild 2.6) bietet nun die Möglichkeit, die in Abschnitt 3.3 beschriebene Vorgehensweise in 5 Schritten interaktiv durchzuführen. Der Entwurfseditor bietet dabei die Möglichkeit, sowohl statische Aspekte (Layout) als auch dynamische Aspekte (Interaktion, Dialogführung) bereits während des Entwurfs testen und bewerten zu können.

4.2.1 Beschreibung der Werkzeugoberfläche

Die Benutzeroberfläche des Entwurfseditors zeigt eine Rollbalken-Menüleiste am oberen Bildrand und eine vertikale Leiste mit verschiedenen Tasten auf der linken Seite (Farbbild 4.1, Anhang B-1). Diese Tasten werden für das Erzeugen neuer Objekte benutzt; mit Hilfe der Menüs kann die Betriebsart des Editors eingestellt werden ("WORKBENCH"), die aktuelle Größe von Objekten verändert ("MODIFY"), die Darstellung eines Objekts bearbeitet ("OBJECT") oder Funktionen wie z.B. Quellkodegenerierung ("SPECIAL") aufgerufen werden.

4.2.1.1 Erzeugen von Objekten und Modifikation der Objekt-Geometrie

Die verschiedenen Objekte des Interfacebaukastens sind als Piktogramme auf den Tasten am linken Bildschirmrand dargestellt (Farbild 4.1). Dies sind im einzelnen: Frame, Slide, Key, Slider (horizontal), Slider (vertikal), Stringpanel, DataView, Text, Icon, und Rect respektive.

Neue Objekte werden mittels der jeweiligen Piktogrammtaste und nachfolgender Positionierung mit dem Mauszeiger erzeugt und zunächst entsprechend ihrer voreingestellten Instanzvariablen am Bildschirm dargestellt. Dieser Vorgang verläuft ähnlich dem Erzeugen graphischer Elemente in bekannten Graphik-Editoren, wie z.B. McDraw™. Das Erzeugen gleichartiger Objekte kann durch Kopieren eines bereits dargestellten Objekts erreicht werden. Im Menü "OBJECT" wird die Funktion "copy" angeboten, um ein Objekt gleicher Höhe und Breite an einer neuen Position darzustellen. Die Funktion "delete" ermöglicht das Löschen eines Objekts.

Die iterative Änderung und Beurteilung der geometrischen Anordnung von Objekten wird durch die Funktionen "move" und "resize" des Menüs "MODIFY" ermöglicht. Die dargestellten Objekte können innerhalb eines Konstruktors verschoben bzw. in der Größe verändert werden. Die Änderung der Größe ist abhängig von der Region, in der das Objekt angeklickt wird (Bild 4.7). Es erscheint dann automatisch ein Cursor, der angibt, welche Objektkanten verschoben werden.

Für die gleichzeitige Bearbeitung mehrerer Objekte können diese zu einer Gruppe zusammengefaßt und z.B. gleichzeitig verschoben werden ("move Group"). Der Aufbau von Symmetrien innerhalb der Gruppe wird von der Funktion "align" unterstützt; sie bietet die Möglichkeit, die Kanten mehrerer Objekte miteinander fluchtend auszurichten.

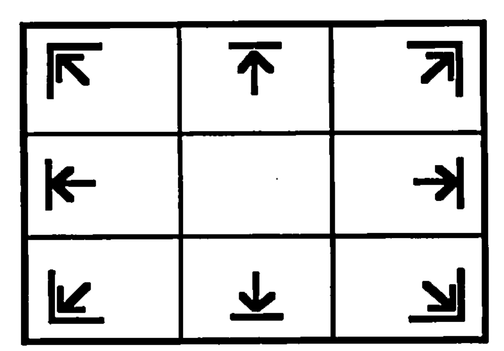


Bild 4.7: Cursor-Formen für die "resize" Funktion.

In Bezug auf die Positionierung kann im Menü "WORKBENCH" eine Rasterweite eingestellt werden, die die Auflösung der pixelgenauen Positionierung einschränkt. Rasterweite 5 bedeutet z.B., daß die jeweiligen Pixelwerte der aktuellen Cursor-Position modulo 5 gerundet werden.

4.2.1.2 Belegung von Objekt-Attributen

Neben der Objektgeometrie besitzt jedes Objekt weitere für das Objekt spezifische Attribute. Die Belegung dieser Attribute wurde in Abschnitt 3.3 als Schritt II der Implementierung eingeführt. Für einen Werteschieber (Slider) muß z.B. der Wertebereich oder das Format der Skalierung eingestellt werden. Die Belegung dieser Attribute wird mit Hilfe eines "Attribut"-Formulars vorgenommen, das durch die Menü-Funktion "attribute" ("OBJECT") aufgerufen werden kann (Bild 4.8).
Auf jedem Formular ist die Option vorgesehen, einem Objekt einen Namen zuzuordnen; dieser Name verbessert die Identifizierbarkeit eines Objekts, da sonst nur eine rechnerinterne Variable das Objekt kennzeichnet. Nach jeder Änderung des Formulars wird das betroffene Objekt entsprechend der eingestellten Werte dargestellt.

Objecttype: SLIDER
Name: rateSlider
Initial State: standby sleep

Valuerange:
From 0. To 10.

Scale: ☒

Format %.0f Font fg-18

Bild 4.8: Attribut-Formular für den Werteschieber.

4.2.1.3 Simulation der Interaktion

Die interaktiv erstellte Benutzerschnittstelle ist jederzeit zur Interaktion bereit, d.h. es handelt sich nicht nur um eine Darstellung der Schnittstelle, sondern bereits um ein interaktives Programm. Die Interaktionselemente (Key, Slider, Stringpanel ...) sind für die Interaktion durch Mauszeiger bzw. Tastatur voreingestellt. Darüber hinaus kann mit dem Editor eine Dialogstruktur definiert werden, wie sie in Abschnitt 3.3 als Schritt IV der Implementierung eingeführt wurde. Die Dialogstruktur wird in einem Attribute-Formular des Konstruktors festgelegt, der die jeweiligen Interaktionsobjekte enthält. Der Dialogmanager kann daraufhin abhängig von der Interaktion die Zustände von Interaktionselementen und Konstruktoren verändern, d.h. schon während des Entwurfs kann der Dialog vollständig simuliert und damit das dynamische Verhalten der Benutzerschnittstelle beurteilt werden.

4.2.1.4 Integration von Bitmap-Editor und Farbeditor

Die Darstellung des Objekts "Icon" sowie die Beschriftung von Tasten ("Key") mit einem Piktogramm erfordern den Aufruf des X-Window Bitmap-Editors. Dieser Editor wird innerhalb der Werkzeugumgebung automatisch aufgerufen, wenn ein "Icon" erzeugt wird oder wenn im "Attribut"-Formular für "Key" ein Piktogramm als Beschriftung ausgewählt wird. In diesem Aufruf wird die Größe des Bitmap entsprechend der Größe des jeweiligen Objekts berücksichtigt. Die interaktive Farbwahl erfolgt mit Hilfe des Werkzeugs CEDI-FT, das in Abschnitt 4.4.4 näher beschrieben ist.

4.2.2 Funktionsweise des Quellkodegenerators

Am Ende einer Sitzung mit dem Entwurfseditor muß die interaktiv am Bildschirm erstellte Benutzerschnittstelle in geeigneter Wiese abgespeichert werden können. Eine besonders effektive Möglichkeit liegt in der Generierung von Quellkode, der nicht nur vom Compiler in ein lauffähiges Programm übersetzt werden kann, sondern gleichzeitig als verständliche Dokumentation für den Entwickler dient.

4.2.2.1 Auswertung der Objekthierarchie

Eine mit dem Interfacebaukasten entworfene Benutzerschnittstelle besitzt einen hierarchischen Aufbau, dessen Baumstruktur ausgehend vom Konstruktor an der Baumwurzel ausgewertet werden kann (vgl. Bild 3.4). Jeder Konstruktor enthält eine Liste mit den zugeordneten Objekten. Dadurch ist es möglich, die Struktur einer Benutzerschnittstelle ausgehend von der Wurzel vollständig zu rekonstruieren.

Mit Hilfe des Entwurfseditors wird die Objektstruktur für eine neue Applikation interaktiv festgelegt und jedes Objekt individuell mit Attributen belegt. Der Quellkodegenerator rekonstruiert nach dem oben genannten Prinzip die Hierarchie, die sich im erzeugten Programmkode in der "childOf:" Beziehung wiederspiegelt (vgl. Abschnitt 3.3.2). Die Attribute, die in Instanzvariablen der Objekte abgelegt sind (z.B. Geometrie), werden durch Nachrichten abgefragt. Darüber hinaus legt der Entwurfseditor zusätzliche Listen an, in denen die Attribute gespeichert sind, die nicht als Instanzvariable verfügbar sind.

Die Erzeugung von leicht lesbarem Kode erfordert die Vergabe von sinnvollen Variablennamen. In Abschnitt 4.2.1.2 wurde bereits die Zuordnung eines Namens für jedes Objekt angesprochen, die neben der rechnerinternen Kennzeichnung auch eine namentliche Identifizierbarkeit ermöglicht. Diesen Namen kann nun der Quellkodegenerator für die Objektvariable benutzen, um verständlichen Quellkode zu erzeugen. Auch die mit diesem Objekt zusammenhängenden Variablen verwenden diesen Namen; so wird z.B. der Variblenname für die geometrische Struktur durch den Namen und den Zusatz "Size" gebildet.

4.2.2.2 Beispiel einer Quellkodegenerierung

Die Benutzerschnittstelle der Infusionspumpen-Kontrolleinheit (Abschnitt 3.3) besitzt eine Teilschnittstelle, die eingeblendet wird, wenn eine neue Rate eingestellt werden soll. Wenn diese Teilschnittstelle interaktiv erzeugt wird, bildet der Kodegenerator für diesen Teil der Schnittstelle z.B. folgenden Quellkode:

```
id adjustment, rateSlider, enterKey;

                    .
XRectangle  adjustmentSize { 0, 170, 700, 95},
            rateSliderSize {10, 20, 570, 30},
             enterKeySize {595, 20, 95, 60};

                    .
adjustment = [Slide create:&adjustmentSize childOf:control];
rateSlider = [Slider create:&rateSliderSize scaleFrom: 0.0
             scaleTo: 10.0 type: HORIZONTAL childOf:adjustment];
enterKey = [Key create:&enterKeySize childOf:adjustment];

                    .
[adjustment setColorTerm:"grey"];
[rateSlider setScale];
[rateSlider setScaleFont: "fg-18"  format:"%.0f"];
[rateSlider setValueWindow];
```

```
[enterKey setLabel:"Enter"];
[enterKey setState:"SLEEP"];
        .
[adjustment setLinkFrom:rateSlider to:enterKey on:STANDBY with:"true"];
        .
[enterKey setFunc: F_Enter];
```

Der Quellkodegenerator bildet also automatisch den in Abschnitt 3.3 besprochenen deskriptiven "interface code" (DIC) für die genannten 5 Schritte der Implementierung einer Applikation. Dieser DIC kann dann direkt durch den Compiler in ein lauffähiges Programm (Maschinenkode) übersetzt werden.

4.2.3 Entwurf und Modifikation von Mehrbildschirm Systemen

In komplexen Überwachungssystemen (z.B. Leitwarten) ist es oft hilfreich bzw. notwendig, die Vielzahl von Informationen getrennt auf mehreren Bildschirmen darstellen zu können. Diese Möglichkeit wurde im Konzept des Interfacebaukastens bereits berücksichtigt. Als Fortführung dieses Konzepts wurde der Entwurfseditor so gestaltet, daß auch schon während des Entwurfs mehrere Bildschirme eingesetzt werden können.

4.2.3.1 Entwurf auf einem seperaten Bildschirm

Der Entwurfseditor bietet die Möglichkeit, das Werkzeug auf einem eigenen Bildschirm optisch völlig getrennt vom Entwurf der Benutzerschnittstelle bedienen zu können. Der Entwurf kann ebenso auf mehrere Bildschirme verteilt erfolgen. Im Menü "WORKBENCH" wird eingestellt, welcher Bildschirm jeweils bearbeitet werden soll.

Für den Entwurf der Benutzerschnittstelle auf separaten Bildschirmen werden jeweils eigenständige Prozesse angelegt, die durch den Nachrichtenaustausch mit dem Werkzeug sukzessive zu einer eigenständigen, interaktiven Benutzerschnittstelle aufgebaut werden. Die Kommunikation zwischen Editor und Schnittstelle basiert auf Mechanismen der "Inter-Client-Communication" von X-Windows. Die Implementierung und das verwendete Protokoll der Kommunikation sind in Nolte (1990) beschrieben.

4.2.3.3 Modifikation laufender Applikationsprogramme

Die Benutzerschnittstelle eines bereits übersetzten und gestarteten Applikationsprogramm kann auch zur Laufzeit durch den Editor weiter modifiziert werden. Dazu wird der Editor vorzugsweise auf einem seperaten Bildschirm benutzt. Die Modifikation wird mit denselben Interaktionssequenzen durchgeführt, die beim Neuentwurf auf einem zweiten Bildschirm vorgenommen werden. Der Eingriff in das laufende Applikationsprogramm erfolgt wieder mit Hilfe der "Inter-Client-Communication", so daß das für den Entwurf festgelegte Kommunikationsprotokoll auch für diesen Bearbeitungsvorgang verwendet werden kann. Die durch den Editor vorgenommenen Änderungen werden intern dokumentiert, so daß auch der den Änderungen entsprechende Quellkode erzeugt werden kann.

4.2.4 Analyse und Bewertung von geometrischen Anordnungen

Die geometrische Anordnung von Informationsgruppen am Bildschirm soll dem Ziel dienen, den Benutzer in seinen Fähigkeiten zur Informationsaufnahme und -verarbeitung zu unterstützen, um so kurze Reaktionszeiten und eine geringe Fehlerrate zu erreichen. Qualitative Richtlinien bezüglich dieser Zielsetzung bestehen in Aussagen wie:

- das Format soll gut strukturiert sein,
- der Bildschirm darf nicht überladen sein,
- die Anordnung soll ästhetisch ansprechend wirken.

Diese Aussagen betreffen die subjektive Beurteilung und können nur durch Benutzerbefragung ausgewertet werden. Quantitativ bestimmbare Faktoren hingegen könnten schon beim Entwurf die objektive Bewertung des geometrischen Layouts unterstützen. Diese Thematik wurde bislang ausschließlich am Beispiel von alphanumerischen Displays bearbeitet (Tullis 1983, Streveler und Wasserman 1984). Die in diesen Arbeiten untersuchten Faktoren sind:

1. Räumliche Dichte
   a. global: Anzahl der Buchstaben auf dem Bildschirm in Bezug auf die vorhandenen Leerzeichen.
   b. lokal: Buchstaben zu Leerzeichen Verhältnis in einer 5 Grad Sehwinkel umfassenden Fläche.
2. Anzahl von Informationsgruppen:
   Zeichen, die durch weniger als zwei Leerzeichen getrennt sind, werden zu einer Gruppe zusammengefaßt.
3. Größe der Informationsgruppen:
   Fläche die eine Gruppe einnimmt.

4. Anzahl der Informationsitems:
   Anzahl der dargestellten Wörter und Zahlen.
5. Layout Komplexität:
   Informationsgehalt in Bit, der in horizontaler und vertikaler Richtung getrennt angibt, wie gut die einzelnen Informationsitems zueinander ausgerichtet sind.
6. Symmetrie und Balance:
   Flächenschwerpunkt der Gruppen in Bezug auf den Mittelpunkt.

Mit Hilfe dieser Faktoren wird eine von der Bedeutung der Applikation (=semantischen Ebene) unabhängige Analyse (=syntaktische Analyse) vorgenommen, so daß diese Berechnungen computerunterstützt auf unterschiedliche Benutzerschnittstellen angewendet werden können. Tullis (1988) konnte in seinen Experimenten nachweisen, daß die o.g. Faktoren z.T. miteinander korrelieren. Für die Suchzeit von Dateneinträgen auf einem Display waren die Größe und Anzahl von Gruppen die maßgebliche Größe, während die subjektive Einschätzung der Güte eines Bildschirmentwurfs am stärksten mit der Layout-Komplexität korrelierte.

Dieser Ansatz soll nun von alphanumerischen Tabellen auf graphische Darstellungen übertragen werden. Eine mit dem Interfacebaukasten entwickelte graphische Benutzerschnittstelle besteht aus Interaktions- und Beschriftungsobjekten, die mittels Konstruktoren hierarchisch strukturiert sind. Dadurch wird im Gegensatz zu alphanumerischen Displays eine geeignete Gruppenbildung der Information bereits durch Umsetzung der Aufgabenanalyse in eine Konstruktorenstruktur erreicht. Die Suchzeit für Dateneinträge ist in graphischen Displays nicht unbedingt vorrangig, da häufig ein Gesamtbild eines Systems vermittelt werden soll (Monitoring). Daher wird nun der Faktor der "Layout-Komplexität" einer geometrischen Anordnung näher betrachtet. Gleichzeitig zeigt der folgende Ansatz am Beispiel der Layout-Komplexität, wie sich eine bekannte Bewertungsmethode an den Entwurfsprozeß für Benutzerschnittstellen anpassen und in ein interaktives Werkzeug der Entwicklungsumgebung integrieren läßt. Eine grundlegende Voraussetzung für die Integration dieser Bewertungsmethode ist die Möglichkeit der Auswertung einer vorliegenden Objekthierarchie von Bausteinen des Interfacebaukastens und der Zugang zu deren Rechteckgeometrie (Ursprung und Größe) über die gespeicherten Instanzvariablen.

Die Berechnung der Layout-Komplexität geht auf einen Ansatz von Bonsiepe (1968) zurück, der diese Größe für die Beurteilung der Gestaltung von Katalogseiten verwendete. Er unterscheidet die Größe und die Positionierung von Informationsblöcken, die sich durch Rechtecke einfassen lassen. Länge, Höhe, x, und y-Ursprung bilden jeweils eine eigene Komplexitätsgröße. Die Hypothese ist, daß die Ordnung einer Darstellung umso größer ist, je weniger verschiedene Klassen einer Größe

vorkommen und je mehr Elemente zu einer Klasse gehören (vgl. Gestaltgesetze). Das bedeutet z.B., daß die Kanten von verschiedenen Objekten soweit möglich auf einer fluchtenden Linie liegen sollen. Die Layout-Komplexität C wurde von Bonsiepe in Anlehnung an die Shannon-Formel definiert:

$$C = -N \sum_{i=1}^{i=n} p_i * ld(p_i) \qquad (4.3)$$

N ist die Anzahl der Rechtecke, n die Anzahl der verschiedenen Klassen und p_i die Wahrscheinlichkeit für ihr Auftreten. Zur Erläuterung ist in Bild 4.10 die Geometrie für einen Entwurf eines Teils der Infusionspumpen-Kontrolleinheit (vgl. Abschnitt 3.3) abgebildet. Es sind 7 Objekte dargestellt; 3 der Objekte haben denselben x-Ursprung und gehören daher zu einer Klasse. Insgesamt gibt es 5 verschiedene Klassen von x-Ursprungskoordinaten, so daß die Komplexität wie folgt errechnet wird:

$$C(X) = - 7 * (3/7*ld(3/7) + 4*1/7*ld(1/7)) = 14.9 \text{ bit}$$

Durch Anwendung von Gleichung 4.3 auf die anderen Komplexitätsgrößen ergibt sich:

Y-Ursprung: 5 Klassen	C(Y) = 15.7 bit
Länge: 5 Klassen	C(Länge) = 14.9 bit
Höhe: 5 Klassen	C(Höhe) = 14.9 bit
	60.4 bit

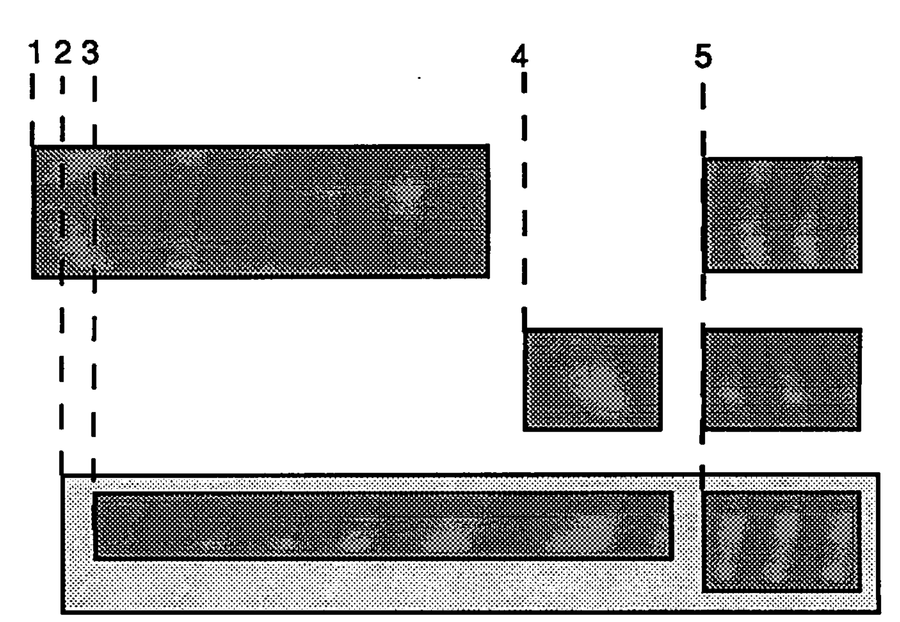


Bild 4.10: Mögliche Geometrie der Infusionspumpen-Kontrolleinheit.

Die Komplexität der Klassen x- und y-Ursprung kann dadurch verringert werden, daß die Rechtecke - wenn möglich - fluchtende Kanten besitzen. Die Komplexität von Länge und Höhe verringert sich, wenn viele Rechtecke dieselbe Höhe bzw. Länge besitzen. Diese Prinzipien sind in Bild 4.11 verwirklicht worden. Die Komplexität dieses Entwurfs berechnet sich zu:

X-Ursprung: 4 Klassen	C(X) = 12.9 bit
Y-Ursprung: 4 Klassen	C(Y) = 13.7 bit
Länge: 4 Klassen	C(Länge) = 11.8 bit
Höhe: 3 Klassen	C(Höhe) = 8.0 bit
	46.4 bit

Die Summe der Komplexitäten konnte also um 14 bit, d.h. um 23 % verringert werden.

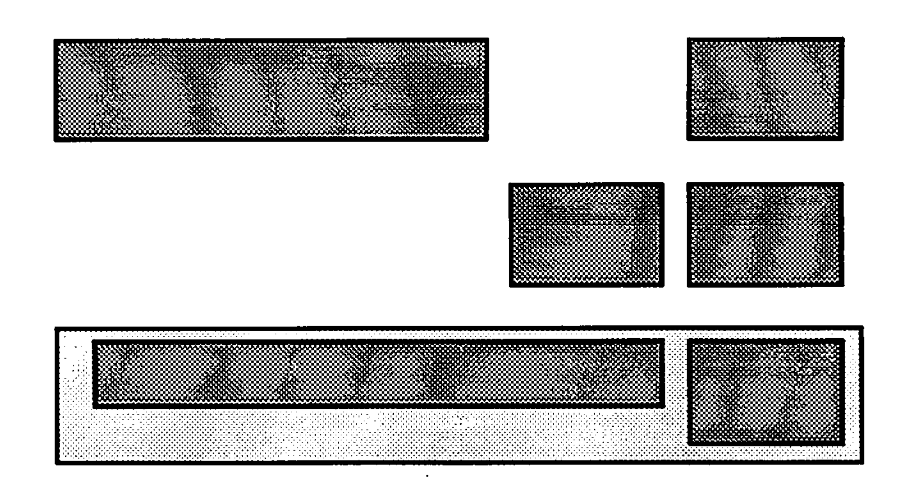


Bild 4.11: Verbesserte Geometrie der Infusionspumpen-Kontrolleinheit.

Eine detaillierte oder abschließende Bewertung der Schnittstellengeometrie ist mit Hilfe der berechneten Komplexität nicht möglich, weil der semantische Bezug zur Aufgabenanforderung der Schnittstelle nicht berücksichtigt werden kann. Die Berechnung der Komplexitäten bietet eine Unterstützung während des Entwurfs, indem sehr schnell vergleichende Bewertungen an einer Schnittstelle durchgeführt werden können. Dabei kann sowohl die Komplexität innerhalb einzelner Konstruktoren als auch die auf den ganzen Bildschirm bezogene Komplexität bestimmt werden. Die Verringerung der Komplexität wird durch die Editor-Funktionen "copy" und "align" vereinfacht, indem fluchtende Kanten bzw. Objekte gleicher Größe rechnergestützt erzeugt werden.

Damit sind die zu Beginn dieses Abschnitts (S. 58) formulierten Anforderungen an ein interaktives Entwurfswerkzeug realisiert und im einzelnen beschrieben worden. Die methodische Vorgehensweise und die technischen Voraussetzungen für einen schnellen ersten Entwurf einer Benutzerschnittstelle stehen somit zur Verfügung.

Nachfolgend werden nun zwei wichtige Teilaspekte der visuellen Informationsdarstellung betrachtet, die Gestaltung von Piktogrammen und die Farbgestaltung.

4.3 Entwicklung von Piktogrammen

Piktogramme haben besonders durch öffentliche Leitsysteme (z.B. auf Bahnhöfen oder Flughäfen) und die Kennzeichnung von Sportarten (Olympische Spiele) Bekanntheit erlangt. In den letzten Jahren gewinnen sie zunehmend Bedeutung bei der Gestaltung von Benutzerschnittstellen im Computer-Bereich. Der Schwerpunkt liegt dabei auf der Bedienung von CAD- oder CAD-ähnlichen Systemen (z.B. GEM-Draw, McPaint) und in der Bürokommunikation ("Desktop-Metapher"). Bei geeigneter Gestaltung werden Piktogrammen folgende Eigenschaften zugeordnet (Staufer 1987, Lodding 1983, Arend und Wandmacher 1988):

- international verständlich, unabhängig von Bildung und Ausbildungsniveau,
- geringer Platzbedarf,
- schnelles Erlernen eines Informationssystems,
- positive Anfangsmotivation des Benutzers durch höhere subjektive Attraktivität der Schnittstelle,
- andauernde Akzeptanz eines neueingeführten Systems,
- größere kognitive Verarbeitungstiefe gegenüber Text durch Nutzung eines adäquaten mentalen Modells,
- kurze Reaktionszeiten bei der Interaktion.

Diese positiven Eigenschaften von Piktogrammen müssen für neue, noch nicht standardisierte Piktogramme erst erreicht werden. Dieser Gestaltungsprozeß kann durch Werkzeuge und Methoden für den Entwurf und die Bewertung unterstützt werden. Dabei kann die Möglichkeit der Benutzerbeteiligung effektive Unterstützung bieten. Nachfolgend werden zuerst Grundlagen aus dem Bereich der kognitiven Verarbeitung von Piktogrammen aufgeführt, um dann auf den Entwurfsprozeß und die Gestaltung neuer Piktogramme für den medizinischen Bereich einzugehen.

4.3.1 Kognitive Verarbeitung von Piktogrammen

Verbale Information in Form geschriebenen Textes ist ein visueller Reiz, der in einen phonetischen Kode verwandelt wird. Es werden daher sowohl das visuelle (rechte Hemisphäre) als auch das auditive Kurzzeitgedächtnis (linke Hemisphäre) belastet. Die seriell arbeitende, analytische Informationsverarbeitung der linken Hemisphäre kann entlastet werden, wenn die Text-Information als Bild (Piktogramm) übermittelt werden kann. Der Nutzen von Piktogrammen liegt darin, die Informationsaufnahme zu erleichtern, indem Systeminformation und Benutzerkommandos in bildhafter Form repräsentiert werden (Lodding 1983). Die "formale Distanz" (Huchins et al. 1986) zur

Information wird geringer, d.h. die Beziehung zwischen der Bedeutung und der physischen Form eines Objekts wird enger.

4.3.1.1 Der Verarbeitungsprozeß

Die Betrachtung des an sich sehr komplexen Vorganges der kognitiven Verarbeitung wird für die weiteren Betrachtungen auf die Unterscheidung von drei Stufen der Verarbeitung (Bild 4.12) bei der Identifikation von Piktogrammen reduziert (Staufer 1987).

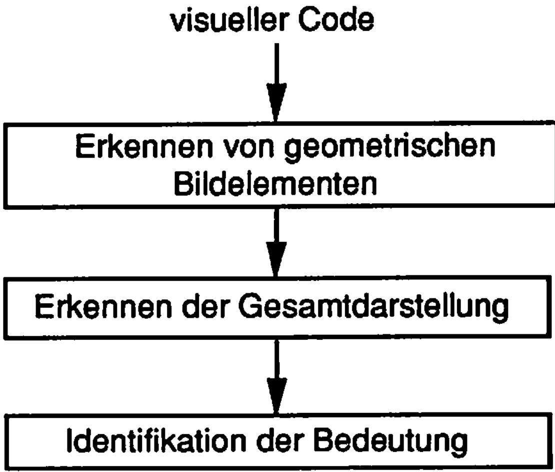


Bild 4.12: Verarbeitungsprozeß bei Piktogrammen.

Die Wahrnehmung der Gesamtdarstellung erfolgt meist sehr schnell durch die parallele Verarbeitung in der rechten Hirn-Hemisphäre. Die Prozesse auf semantischer Ebene (Identifikation) sind hingegen zeitaufwendiger, da hierbei verstärkt auf kognitive Ressourcen zugegriffen werden muß. Eine Studie von Lees und Farman (1970) zeigte, daß bildhafte Darstellungen oft als "gut erkennbar" eingestuft wurden, die Bedeutung (Semantik) jedoch nicht näher beschrieben werden konnte. Die semantische Ebene ist aber entscheidend für die Behaltensleistung; diese kognitive Leistung korrespondiert nach einem Modell von Craik und Lockhart (1972) mit der Verarbeitungstiefe, d.h. der Tiefe der semantischen und kognitiven Analyse. Die für Piktogramme relevanten Faktoren sind:

- Einwirkzeit,
- Menge des präsentierten Materials,
- Bekanntheitsgrad und Kompatibilität,
- Bedeutungshaltigkeit.

Die letzten beiden Punkte kennzeichnen die Überlegenheit der Piktogramme gegenüber Text, weil das kognitive Modell des Betrachters von einem Objekt die größere Kompatibilität und Bedeutungshaltigkeit zu einem Piktogrammen haben kann.

4.3.1.2 Schwierigkeiten beim Entwurf von Piktogrammen

Die Vorteile hinsichtlich der größeren Kompatibilität und Bedeutungshaltigkeit von Piktogrammen werden in folgenden Überlegungen eingeschränkt (Arend 1988, Myers 1984):

- Es gibt nicht für jede Operation eine offensichtlich geeignete piktografische Darstellung. Eine verständliche Darstellung der "UNDO"-Funktion konnte sich bislang nicht durchsetzen.
- Die Unterscheidbarkeit mehrerer Begriffe ist nur gewährleistet, wenn auch die Piktogramme leicht unterscheidbar gestaltet werden.
- Piktogramme, die das Erlernen eines Systems erleichtern, können den fortgeschrittenen Benutzer möglicherweise behindern.
- Piktogramme müssen (mehr oder weniger) gelernt werden.
- Es ist wenig bekannt, wie Piktogramme optimal zu gestalten sind.
- Es gibt nur wenige verwendbare Standards.

Diese Gründe zeigen, daß der Entwurf eines Piktogramms nicht im "ersten Wurf" gelingen wird, sondern auch hier ein Prototyping Prozeß notwendig wird. Die willkürliche Gestaltung von Piktogrammen nach der Vorstellung eines Entwicklers kann dazu führen, daß ein Piktogramm schwer verständlich oder mißverständlich ist. Während des Prototyping Prozesses soll daher ein adäquates Modell der Vorstellung des Benutzers gefunden werden, das ein ausreichendes Maß an Kompatibilität und Bedeutungshaltigkeit erreicht.

4.3.1.3 Der Gestaltungsspielraum der Darstellung - Klasseneinteilung von Piktogrammen

Bilder im Sinne einer photographischen Abbildung haben für den Menschen den höchsten Grad von Anschaulichkeit, Zeichen (z.B. Buchstaben) oder Symbole den geringsten. Die Darstellung eines Piktogramms bewegt sich innerhalb dieser Grenzen und kann entweder mehr zum Abstrakten oder zum Photografischen hin tendieren. Dieser Spielraum ist in Bild 4.13 an Piktogrammen für ein Telefon verdeutlicht.

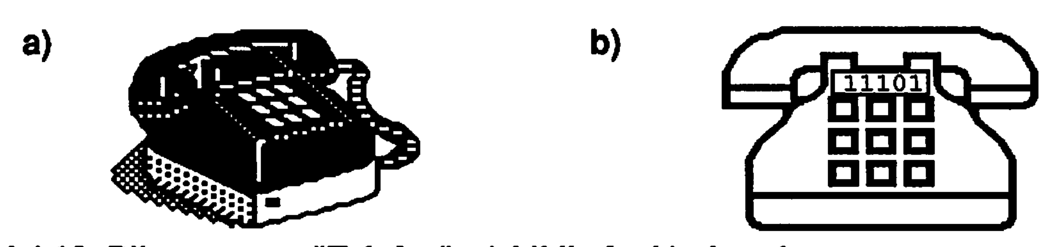


Bild 4.13: Piktogramme "Telefon" a) bildhaft, b) abstrakt.

Neben diesen "objektbezogenen" Darstellungen gibt es auch Piktogramme, die nicht direkt das Dargestellte ausdrücken sollen, sondern das dargestellte Objekt steht für einen assoziativen Begriff oder einen bestimmten Aspekt eines abstrakten Konzepts. Diese Klasse von Piktogrammen soll daher als "assoziativ" bezeichnet werden (Bild 4.14).

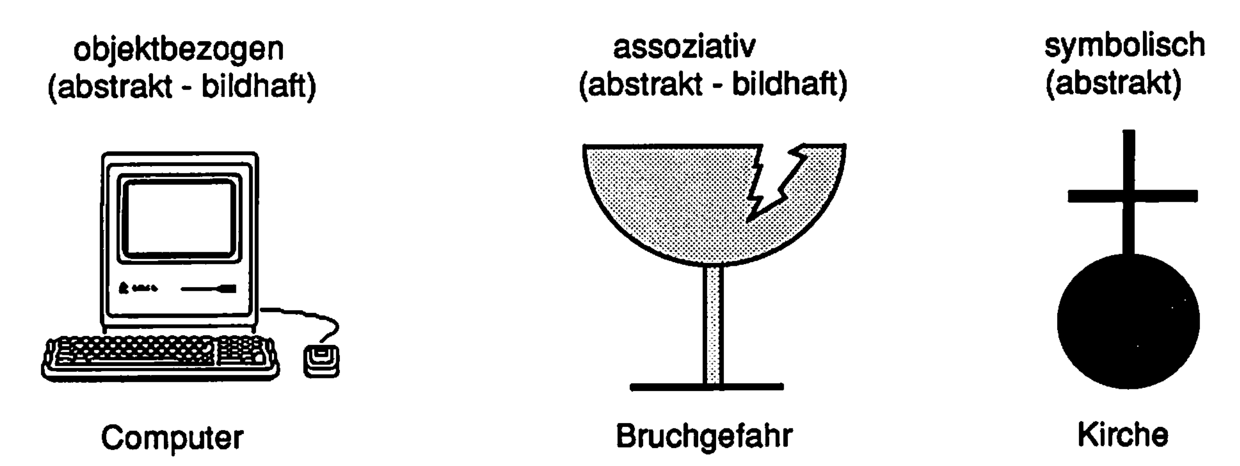


Bild 4.14: Klasseneinteilung von Piktogrammen

Für viele Begriffe existiert keine abbildhafte Darstellung, so daß hier die Darstellung eines assoziativ verknüpften Objekts möglich ist. Assoziative Zusammenhänge sind oft nicht unmittelbar erkennbar und müssen erst gelernt werden (z.B. Assoziation "Post" für das dargestellte Horn in Bild 4.15). In Anlehnung an das Modell in Bild 4.12 ist festzuhalten, daß die Stufe der Identifikation für assoziative Piktogramme größere kognitive Anstrengung erfordert. Die dritte der in Bild 4.14 dargestellten Klassen bilden symbolische Piktogramme und indexikalische Zeichen. Der Bedeutungsinhalt dieser Piktogramme muß immer erst erlernt werden. Der Entwurf neuer Symbole für piktografische Schnittstellen für nichttechnische Benutzer ist daher nicht zweckmäßig, die Nutzung schon bekannter Symbole hingegen möglich. Erfahrungsgemäß werden sowohl objektbezogene als auch assoziative Piktogramme während des Entwicklungsprozesses ausgehend von einer zunächst bildhaften Darstellung zunehmend abstrahiert. Dadurch wird die visuelle Komplexität verringert und somit eine schnellere Verarbeitung des Reizes möglich; die Reaktionszeit auf den visuellen Reiz wird kürzer (Arend 1988). Diese Abstraktion entwickelt sich meist schrittweise - oft über viele Jahre hinweg, indem typische Merkmale extrahiert und abstrahiert werden (Bild 4.15).

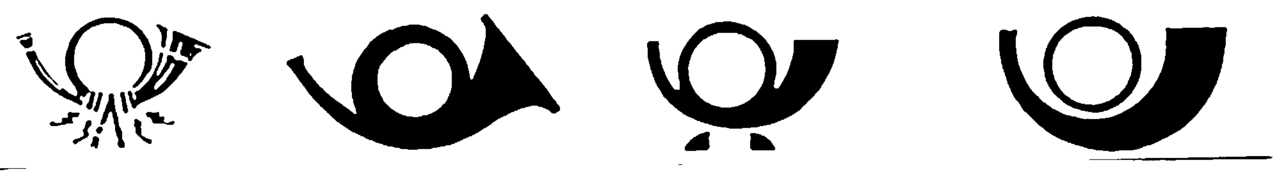


Bild 4.15: Historische Entwicklung des Piktogramms "Post" (aus: Staufer 1987).

Wenn bereits bei der Neueinführung eines Piktogramms ein hoher Abstraktionsgrad verwendet wird, muß mit Interpretationsschwierigkeiten seitens der Benutzer und einer geringen Akzeptanz gerechnet werden. Es besteht daher ein Trade-Off (wechselseitiger, konkurrierender Austausch) zwischen einem hohen Grad der Abstraktion, d.h. Reduktion auf wenige charakteristische Merkmale, die eine schnelle Verarbeitung ermöglichen, und einer detailierten Darstellung, die selbsterklärend eine spontane Interpretierbarkeit von Piktogrammen erlaubt. Um sowohl prägnante und schnell erfaßbare als auch leicht interpretierbare Piktogramme zu entwerfen, ist daher ein Kompromiß erforderlich, über dessen Erfolg nur der Benutzer selbst entscheiden kann.

4.3.2 Systematik des Entwicklungsprozesses

4.3.2.1 Prinzipien für die Piktogrammentwicklung

Ein Piktogramm soll typisch für ein Objekt, eine Aktion oder abstrakte Vorstellung sein, d.h. es soll relevante Merkmale aufweisen, die sich in das mentale Modell des Benutzers einfügen. Dabei können die in Kapitel 4.1 aufgeführten Prinzipien (z.B. Gestaltsgesetze) eingesetzt werden. Die Gestaltungsziele sind abhängig von der Phase innerhalb des Gesamtprozesses unterschiedlich.

Bei Neueinführung:
- ein bildhafter Eindruck des Objekts soll die spontane Erkennbarkeit ermöglichen.

Bei der Standardisierung bekannter Piktogramme:
- geringe visuelle Komplexität,
- nur grobe Merkmalselemente verwenden, d.h. Bildhaftigkeit, funktionale Information und Objektbezogenheit spielen keine oder eine negative Rolle (Arend 1988).

Zusätzlich müssen beim Entwurf eines Satzes von Piktogrammen die Beziehungen der Piktogramme untereinander berücksichtigt werden:
- eine hohe visuelle Unterscheidbarkeit von Piktogrammen innerhalb eines Satzes soll Verwechselungen zwischen den einzelnen Piktogrammen verhindern,
- ein einheitlicher Darstellungsstil der Piktogramme soll eingehalten werden, sofern sie sich auf verwandte Objekte oder Aktionen beziehen. Dabei muß abgewogen werden, ob die Konsistenz eines Satzes durch bestimmte -einzeln als optimal bewertete- Piktogramme gefährdet wird. Es ist besser, einen Satz mittelmäßiger Piktogramme zu haben als vereinzelt sehr gute und sehr schlechte.

4.3.2.2 Produktion von Piktogrammentwürfen

Für neue Piktogrammentwürfe sind Gestaltungsideen erforderlich, die durch folgende Vorgehensweisen gefunden werden können:

- Verbale Beschreibung des Begriffs (semantischer Inhalt), eventuell durch Interviews mit den Benutzern,
- Ermittlung des semantischen Umfelds (Assoziationen),
- Beobachtungen im Arbeitsumfeld der Benutzergruppe,
- Anfertigung und Diskussion von Skizzen.

Als Ergebnis entstehen erste Entwürfe, wobei zu jedem Begriff mindestens 2 Alternativen zur Verfügung stehen sollten. Die Entwürfe können dann zunächst intern von verschiedenen Systementwicklern bewertet und verbessert werden, bevor sie dann der Evaluierung durch typische Systembenutzer zugeführt werden.

4.3.2.3 Methoden der Evaluierung von Piktogrammen

Eine Evaluierung ist besonders effektiv, wenn sie in Zusammenarbeit mit den zukünftigen Benutzern durchgeführt wird. Folgende Qualitätskriterien können zur Beurteilung herangezogen werden:

1. Erkennen: ist der Bildinhalt für den Betrachter erkennbar?
2. Interpretieren: wird der intendierte Bedeutungsinhalt verstanden, d.h. der richtige Begriff zugeordnet? (Charakteristik, Typikalität)
3. Eindeutigkeit: unterscheidet sich ein Piktogramm ausreichend von den anderen Piktogrammen, d.h. kommt nur ein Piktogramm für einen Begriff in Frage? (Verwechselbarkeit)
4. Erlernbarkeit und Behaltbarkeit der Piktogramme.
5. Ästhetischer Reiz der Darstellung.

Für die Überprüfung der genannten Qualitätskriterien wurden u.a. folgende Methoden vorgeschlagen (Staufer 1987):

1. Wiedererkennungsmethode oder Verständnismethode: Dem Benutzer werden die Piktogramme einzeln präsentiert, woraufhin dieser einen dazu passenden Begriff nennt.
2. Zuordnungsmethode: Alle Piktogramme und dazugehörigen Begriffe werden präsentiert, woraufhin der Benutzer jedem Piktogramm einen der vorgegebenen Begriffe zuordnet.
3. Auswahl-Antwort Verfahren: Es wird ein Begriff genannt und ein Piktogramm gezeigt. Der Benutzer entscheidet, ob das Piktogramm dem Begriff entspricht. Als Meßgrößen dienen Reaktionszeit und Fehlerrate.
4. Lesbarkeitsentfernung: Es wird die weiteste Entfernung bestimmt, in der der Benutzer die Bedeutung des Piktogramms noch benennen kann.

Der gesamte Ablauf der Piktogrammentwicklung ist in Bild 4.16 zusammengefaßt:

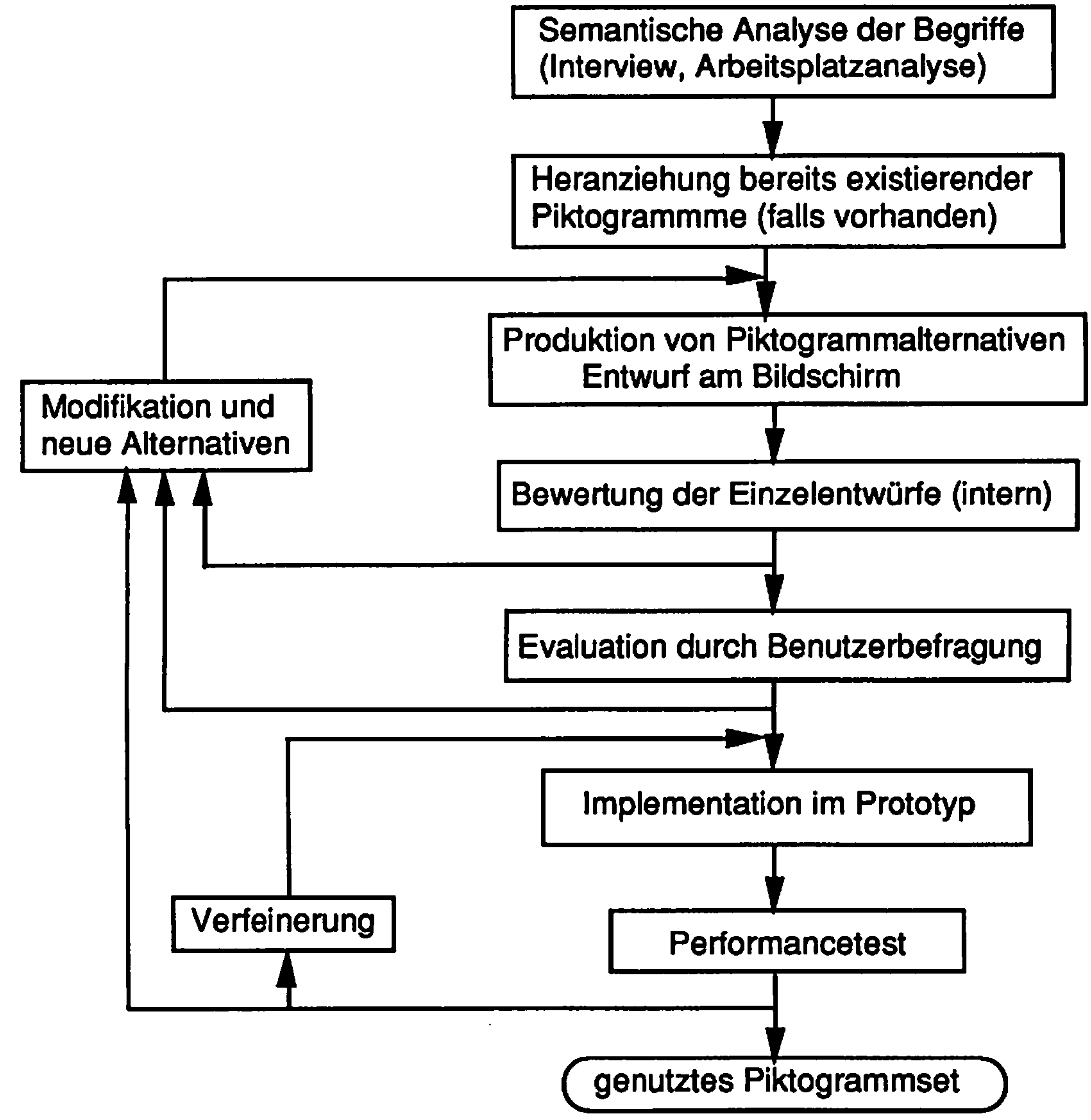


Bild 4.16: Ablauf einer Piktogrammentwicklung.

Die Planung, Durchführung und Auswertung einer Bewertungsstudie von Entwürfen kann nach Methoden aus der experimentellen Psychologie erfolgen. Hier sind an erster Stelle Rating-Experimente zu nennen bzw. für die Wahrnehmung isolierter Reize die Berechnung von ROC-Kurven. Eine detailierte Übersicht über verschiedene Bewertungsverfahren wird in Meister (1986) gegeben.

4.3.3 Entwicklung und Bewertung von Piktogrammen - Fallstudie

Die Verwendung piktografischer Darstellungen in der Medizin ist auf einigen Geräten bereits ansatzweise zu beobachten, jedoch ist ein Trend zur Standardisierung noch nicht erkennbar. Die folgende Fallstudie beschreibt die Entwicklung und Bewertung von Piktogrammen für ein komplexes System zur Patientenüberwachung. In diesem System war die Visualisierung von Hinweisen zur Entscheidungsunterstützung eine schwierige Aufgabe, so daß eine Lösung mit Hilfe von Piktogrammen untersucht wurde.

4.3.3.1 Zieldefinition und Vorgehensweise

Als Beispiel für die Verwendung von Piktogrammen wird ein Teil der Benutzeroberfläche eines Anästhesie-Entscheidungs-Unterstützungs Systems beschrieben. Das Konzept der Wissensbasis für das Anästhesie-Entscheidungs-Unterstützungs System AES-2 beschreiben Schecke et al. (1988). Das Gesamtkonzept der Benutzerschnittstelle wird in Kapitel 6 beschrieben.

Für die Benutzeroberfläche des AES-2 war die Aufgabe zu lösen, 5 Zustandsgrößen des Patienten und deren aktuellen Trend visuell zu kodieren. Dies sind im einzelnen folgende Begriffe:

> Kontraktilität, Vasotonus, Intravasalvolumen, Herzfrequenz, Narkosetiefe; Trend fallend, Trend steigend, Trend konstant.

Weiterhin sollten Piktogramme den Inhalt von einzelnen Informationsfenstern kennzeichnen und die vier physiologischen Teilsysteme des Menschen repräsentieren können:

> Monitoringdaten, Patientendaten, Therapieempfehlung, Therapieerklärung, Therapieliste; kardiovaskuläres System, Nierensystem, respiratorisches System, Zentralnervensystem.

Zu jedem Begriff wurden zunächst 2 Piktogrammalternativen entwickelt und gemeinsam mit zwei Ärzten[1] verfeinert. Eine Abbildung der 32 Piktogramme findet sich im Anhang. Im Rahmen einer heuristischen Pilotstudie wurden diese Entwürfe vom medizinischen Personal der Klinik für Anästhesiologie der RWTH bewertet. Die Befragung wurde mittels Fragebogen durchgeführt, um möglichst viele Personen in die Bewertung miteinzubeziehen. Der Fragebogen wurde bei eigener Anwesenheit mit den Teilnehmern (n=30) in deren Arbeitsumfeld (OP) ausgefüllt, so daß das Problem

[1] Diese Ärzte waren aufgrund langjähriger Zusammenarbeit zwischen dem Helmholtz-Institut für Biomedizinische Technik (Direktor: Prof. Dr. rer. nat. G. Rau) und der Klinik für Anästhesiologie (Direktor: Prof. Dr. med. G. Kalff) mit dem Konzept des AES-2 vertraut.

eines geringen Rücklaufs von Fragebögen umgangen wurde und gleichzeitig der Kontakt mit dem Benutzer am Arbeitsplatz (reale Umgebung) neue Impulse bringen konnte.

4.3.3.2 Entwurf eines Fragebogens

Die in Kapitel 4.3.2.3 aufgeführten Qualitätskriterien Erkennbarkeit, Interpretierbarkeit, Charakteristik (=Typikalität), Verwechselbarkeit und Ästhetik sollten im Fragebogen überprüft werden. Da die Teilnehmer während der Arbeitszeit am Arbeitsplatz befragt wurden, mußte die mentale Belastung durch die Befragung möglichst gering bleiben, d.h. der Fragebogen leicht verständlich und schnell zu bearbeiten sein. Zur Überprüfung dieser Anforderung wurde eine Vorstudie mit 5 Medizinstudenten durchgeführt, der die Verständlichkeit der Fragen und die Bearbeitungsdauer (Vorgabe maximal 20 min.) untersuchte. Diese Vorstudie führte zur Modifikation des Fragebogens sowie zur Verfeinerung einiger Piktogramme.

In der überarbeiteten Version (Abbildung im Anhang) besitzt der Fragbogen eine kurze Einleitung, die den Teilnehmer(in) über die Motivation für diesen Fragebogen informiert; es werden einige personenbezogene Daten (Alter, Berufsgruppe, ...) erhoben und die Frage nach der persönlichen Einstellung zur Verwendung von Piktogrammen im medizinischen Bereich gestellt. Der Hauptteil des Fragebogens gliedert sich in drei Teile:

Teil I:

Der Abbildung eines Piktogramms soll vom Teilnehmer spontan ein passender Begriff (=Identifikation) zugeordnet und auf einer freien Zeile eingetragen werden. Zusätzlich soll mittels einer 7-stufigen Rating-Skala die Erkennbarkeit des Bildinhalts und der ästhetische Reiz (unipolare Adjektivpaare) eingestuft werden. In diesem Teil sind die für die Piktogrammme vorgesehenen Begriffe den Teilnehmern unbekannt. Es wird nur eine Hälfte (16er Set) der Gesamtkollektion dargeboten, d.h. nur eine Alternative für jeden Begriff. Es entstehen dadurch 2 Gruppen von Fragebögen, um eine Bewertung von jedem der 32 Piktogramme in Teil I zu erhalten.

Teil II:

Ensprechend der Zuordnungsmethode wird dem Teilneher die gesamte Kollektion (32er Set) und die Liste der 16 vorgesehenen Begriffe dargeboten. Als Ergebnis erhält man den Anteil richtiger Zuordnungen und die Art von Verwechslungen.

Teil III:
Wieder wird eine Hälfte (16er Set) der Gesamtkollektion dargeboten, diesmal mit der Angabe des zugeordneten Begriffs. Auf einer Rating-Skala wird die Charakteristik des Piktogramms (Typikalität) eingeschätzt. Die in Teil III dargebotene Teilkollektion ist komplementär zu der in Teil I, damit die Typikalität unabhängig von einer möglichen Falsch-Zuordnung in Teil I bewertet werden kann. Durch die Bildung von 2 Gruppen von Fragebögen wird jede Teilkollektion gleichhäufig in Teil I und III dargeboten.

4.3.3.3 Methodik der Auswertung und Ergebnisse
Für die verschiedenen Rating-Größen wurden Mittelwerte und Standardabweichungen berechnet. Für den Vergleich zweier Alternativen wurde der T-Test verwendet, um die Signifikanz einer Differenz zu bestimmen. Das Signifikanzniveau einer Abweichung zweier Mittelwerte wird für den gewählten Stichprobenumfang wie folgt bestimmt:
Signifikanz auf 5% Niveau: $t \geq 2.05$
Signifikanz auf 1% Niveau: $t \geq 2.76$
Signifikanz auf 0.1% Niveau: $t \geq 3.67$
Der Wert von t wird nach folgender Formel berechnet:

$$t = \frac{M_1 - M_2}{\sqrt{\left[\frac{1}{N_1} + \frac{1}{N_2}\right]\left[\frac{\Sigma x_1^2 - \frac{(\Sigma x_1)^2}{N_1} + \Sigma x_2^2 - \frac{(\Sigma x_2)^2}{N_2}}{N_1 + N_2 - 2}\right]}} \qquad (4.7)$$

M: Mittelwert der Ratingwerte einer Alternative
N: Stichprobenumfang
x: einzelner Ratingwert einer Alternative.

Der Anteil richtig identifizierter Piktogramme in Teil I des Fragebogens wird als prozentualer Wert angegeben. Das ISO-Kriterium fordert hier für eine Standardisierung einen Mindestwert von 0.66. Die Auswertung von Teil II zeigt, ob bestimmte Piktogramme und Begriffe häufig miteinander verwechselt werden (geringe Deskriminierbarkeit).

Die Ergebnisse der statistischen Auswertung sind detailliert im Anhang aufgeführt. An dieser Stelle sollen examplarisch eine überraschend positive und eine überraschend negative Bewertung angeführt werden. Der Entwurf für die Kontraktilität in Bild 4.17 a) galt vor der Befragung als originell aber gleichzeitig auch "gewagt". Die Bewertung zeigte dann eindeutig, daß in diesem Fall kein Grund zur Skepsis bestand.

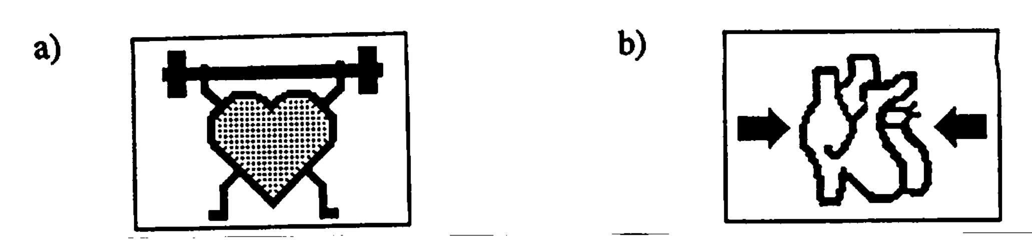


Bild 4.17: Piktogramme für "Kontraktilität".

Bei den beiden Piktogramme für das Intravasalvolumen wurde zunächst Alternative a) (Bild 4.18) gegenüber b) favorisiert. Die Auswertung zeigte, daß beide Piktogramme deutlich schwächer beurteilt wurden als die anderen Piktogramme für Zustandsgrößen; außerdem schnitt Alternative b) besser ab. Die Assoziation zwischen dem Blutvolumen und einer angefüllten Figur ist offensichtlich für den Erstentwurf zu weit gefaßt und nur für "Insider" verständlich.

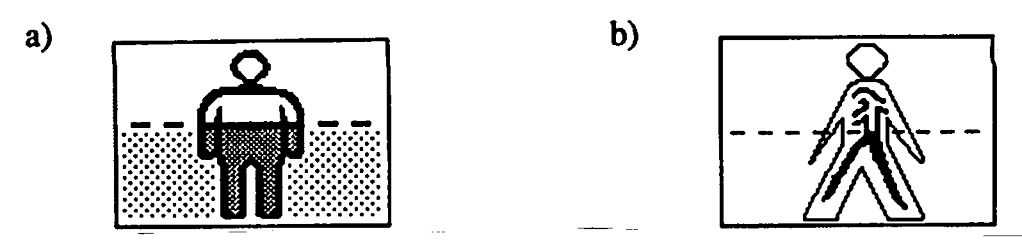


Bild 4.18: Piktogramme für "Intravasalvolumen".

Aus den Ergebnissen konnten folgende neue Gestaltungsprinzipien abgeleitet werden:

- Eine hohe Abstraktion bzw. weitgefaßte Assoziation ist bei Neuentwürfen zu vermeiden.
- Gut verständliche Bildobjekte aus fachfremden Bereichen (Musik, Sport) können die Qualität positiv beeinflußen.
- Die Anzahl der Formelemente pro Piktogramm ist auf höchstens 3 zu beschränken; diese sollen in einem unmittelbaren Zusammenhang stehen. Eine Verdeckung von Bildelementen sollte vermieden werden.
- Die Verwendung von verschiedenen Füllrastern zur Absetzung von Flächen ist sehr effektiv; die Verwendung von Farbe ist hier nicht notwendig.
- Soweit wie möglich sollen Linien im Winkel von 0, 45 und 90 Grad verwendet werden.

Es zeigte sich, daß die Vielzahl der im Fragebogen erhobenen Daten die Gestaltung eines akzeptablen, einheitlichen Satzes von Piktogrammen unterstützt. Beim Vergleich zweier Alternativen unterscheidet sich oft nur eine der Bewertungsgrößen. Die Befragung von insgesamt 30 Personen direkt in deren Arbeitsumfeld ist als erste Bewertung der Qualität der Entwürfe und als Einschätzung der Akzeptanz von Piktogrammen in diesem speziellen medizinischen Bereich zu betrachten. Eine endgültige

Bewertung ist erst möglich, wenn Piktogramme in diesem Bereich etabliert und die Benutzer im Umgang mit Piktogrammen erfahren sind (z.B. im Anschluß an einen Feldtest).

Die beschriebene Vorgehensweise für den Entwurf und die Bewertung von Piktogrammen ist ein gangbarer Weg, der sich bei der Gestaltung der Piktogramme für das Anästhesie-Entscheidungs-Unterstützungs System bewährt hat. Es kann zukünftig auch allgemein für vergleichbare Anwendungen eingesetzt werden, um eine zufriedenstellende Lösung bei der Gestaltung neuer Piktogramme zu erreichen.

4.4 Werkzeug zur interaktiven Farbgestaltung

Eine geeignete farbliche Gestaltung auf einem hochauflösenden Bildschirm kann die Darstellung komplexer Information erheblich verbessern. Farbe kann mit unterschiedlicher Zielsetzung zur Kodierung eingesetzt werden (Grimm et al. 1983):
- Erleichterung und Beschleunigung von Suchvorgängen,
- Trennung von Informationskategorien und Klassifizierung,
- Aufmerksamkeitssteuerung,
- Gliederung von Dialogschritten und Transparenz des Dialogablaufs.

Die Bedeutung von Farbe bei Such- und Identifikationsaufgaben wurde erstmals ausführlich von Christ (1975) bearbeitet. In seinen Experimenten und anderen darauf aufbauenden Arbeiten konnte gezeigt werden, daß farbige Suchobjekte - insbesondere auf großen Bildschirmen - schneller gefunden wurden als Objekte, die sich nur in der Form unterschieden. Die kürzeste Suchzeit wurde gemessen, wenn sowohl Farbe als auch Größe redundant zur Kodierung verwendet wurden. Farbe kann besonders effektiv eine Flächensegmentierung verdeutlichen. Dabei erfolgt die Auswertung der Farbinformation parallel zur Perzeption der Form, ohne daß eine zusätzliche kognitive Belastung auftritt (Davidoff 1987).

Untersuchungen von Tullis (1981) zeigten, daß Benutzer eine größere Zufriedenheit bei der Arbeit mit farbigen Bildschirmen gegenüber monochromatischen Bildschirmen äußerten, selbst wenn ihre Leistung unverändert blieb. Es wird daher angenommen, daß die Motivation für die Arbeit mit einem Farbbildschirm größer ist. Die Akzeptanz eines Systems kann durch die Verwendung von Farbe gesteigert werden (Smith 1987). In Experimenten von Pfendler und Widdel (1986) konnte nachgewiesen werden, daß die Aufmerksamkeit bei einer Überwachungsaufgabe beim Einsatz eines farbigen Bildschirms gegenüber einem monochromen anstieg.

Die Unterstützung der Interaktion durch eine Benutzerführung mittels Farbkodierung wurde von Rau und Trispel (1982) eingeführt und bei der Gestaltung eines Anästhesie-Informations-Systems eingesetzt. Eine ausgereifte Lösung beschreibt Klocke (1987).

Heute hat ein Systementwickler bereits auf einer modernen Workstation Farbgraphiksysteme mit einem Farbumfang von 16.7 Millionen Farbwerten zur Verfügung; daher steht die Frage, "ob" Farbe verwendet werden soll, seltener im Mittelpunkt als die Frage, "wie" Farbe am Bildschirm geeignet eingesetzt werden kann, um die genannten Vorteile einer Farbkodierung nutzen zu können. Es ist bekannt, daß eine nicht adäquate Farbgebung die Informationsverarbeitung des Benutzers beeinträchtigen kann. Die

Forschung konzentrierte sich in der Vergangenheit auf die Wirkung von Farbe in alphanumerischen Anzeigen, d.h. hauptsächlich auf die Lesbarkeit von Text (Fukuzumi 1989). Im Bereich der graphischen Farbgestaltung sind jedoch nur grobe, qualitative Richtlinien bekannt, so daß ein Systemdesigner stets auch auf eine heuristische Vorgehensweise angewiesen ist. Bislang fehlen für diese Vorgehensweise geeignete Werkzeuge, die eine sinnvolle Auswahl aus der Vielfalt möglicher Farben unterstützen. Die Effizienz der Werkzeuge kann verbessert werden, wenn durch Anwendung von Farbmetrik auch eine quantitative Farbanalyse ermöglicht wird. Dadurch werden die Voraussetzungen geschaffen, eine farbliche Gestaltung einer Benutzerschnittstelle auch auf farbmetrische Daten gestützt bewerten zu können. Die Farbgestaltung mehrerer untereinander in Beziehung stehender Farbflächen soll durch ein Werkzeug interaktiv variiert werden können, da dies der einzige Weg ist, die subjektive Wahrnehmung simultan dargestellter Farbreize zu berücksichtigen.

In den folgenden Kapiteln wird ausgehend von perzeptiven Grundlagen des Farbensehens und qualitativen, meist empirischen Aussagen zur Farbwirkung eine Verbindung zu metrischen Farbräumen hergestellt. Daraus wird dann das Konzept für ein Werkzeug zur Farbgestaltung abgeleitet und dessen Realisierung erläutert.

4.4.1 Farbwahrnehmung des Menschen

Für die Perzeption von Farbe sind 3 verschiedene Rezeptortypen der Netzhaut verantwortlich, deren Absorptionskurve jeweils im Rot-, Grün- bzw. Blaubereich verläuft; ihre Absorptionsmaxima liegen bei den Wellenlängen 445 nm, 535 nm und 575 nm (Cyanolab, Chlorolab, Erythrolab) (Jameson 1972). Dieser Zusammenhang wurde bereits von Young und Helmholtz postuliert und als "trichromatische" bzw. "Dreikomponeneten" Theorie bezeichnet (v. Helmholtz 1896). Die Gesamtheit der Farben läßt sich demnach in einem dreidimensionalen Raum darstellen und bietet drei Dimensionen der Kodierung. Einen anderen Ansatz begründete Hering in seiner Gegenfarbentheorie (Hering 1920), wo als "Urfarben" Rot, Grün, Blau und Gelb angenommen werden. Die Bezeichnung Gegenfarben hat ihren Ursprung in der Erkenntnis, daß es weder ein "grünliches Rot" noch ein "bläuliches Gelb" gibt.

Beide scheinbar konkurrierenden Systeme haben Gültigkeit, jedoch auf verschiedenen Ebenen der Wahrnehmung (v. Kries 1905). Die Reizaufnahme durch die Rezeptoren erfolgt analog der trichromatischen Theorie, die nachfolgenden Neuronensysteme führen eine Reizverarbeitung durch, die besser mit der Gegenfarbentheorie beschrieben wird. Der rezeptive Prozess bildet die erste Stufe der in Bild 4.19 dargestellten Stufen der Farbwahrnehmung.

Die Rezeptor-Signale werden im Auge vorverarbeitet und in 3 neuronalen Kanälen weitergeleitet. Diese Kanäle führen ein Helligkeitssignal, sowie ein Rot/Grün- und Gelb/Blau-Signal, also jeweils ein Paar sogenannter "Gegenfarben". Im Zentralnervensystem erfolgt die letzte Verarbeitungsstufe der Farbwahrnehmung; die Signale der neuronalen Kanäle werden zu einem bewußten Farbeindruck mit den Komponenten Farbton, Farbsättigung und Helligkeit verarbeitet. Diese Komponenten werden oft für die sprachliche Beschreibung eines Farbreizes herangezogen (z.B. ein "heller, gesättigter Gelbton") und sind daher intuitiv verständlich. Diese kontinuierliche Verarbeitung perzeptiver Information wurde von Mishkin und Appenzeller (1987) so beschrieben:

> "Diese [..] Ergebnisse ließen uns die Theorie aufstellen, daß die visuelle Information entlang der neuronalen Bahn Stück für Stück verarbeitet wird. Die Nervenzellen der Bahn haben gleichsam der Gesichtswelt zugewandte Fenster, die für die aufeinanderfolgenden Stufen immer größer werden, sowohl in ihrer räumlichen Ausdehnung als auch in der Kompliziertheit der Information, die sie eindringen lassen. Die Zellen beantworten zunehmend mehr physikalische Eigenschaften eines Gegenstandes, zu denen Größe, Form, Farbe und Oberflächenstruktur gehören, bis sie schließlich auf den letzten Stufen der unteren Schläfenlappenrinde eine vollständige Repräsentation des Gegenstandes erstellen."

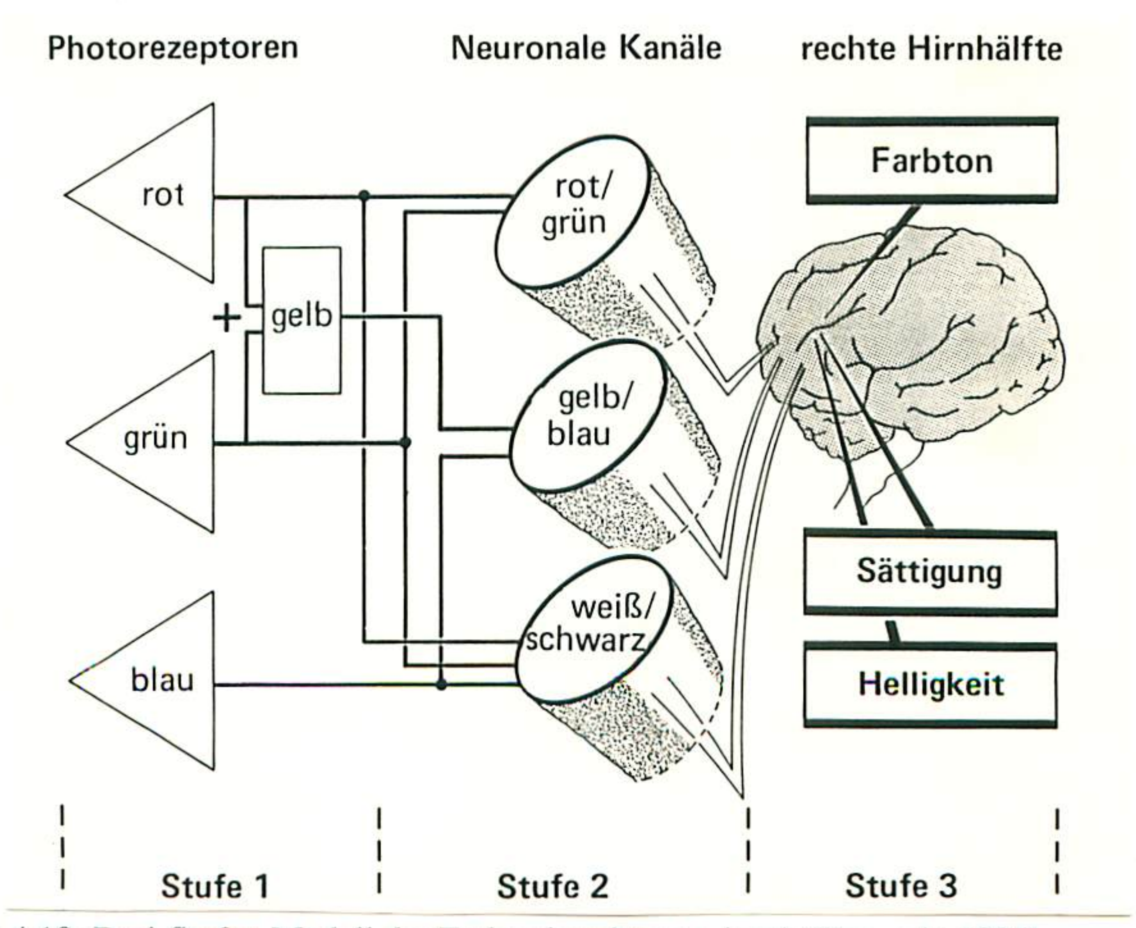


Bild 4.19: Drei-Stufen Modell der Farbwahrnehmung (nach Wyszecki 1986).

Das vorgestellte Modell der Farbwahrnehmung ist grundlegend für die Beschreibung verschiedener Phänomene des Farbensehens und zur Ableitung einiger qualitativer Prinzipien zur Farbgestaltung:

- Bei nebeneinanderliegenden Flächen mit gesättigtem Rot und Blau kann wegen der unterschiedlichen Wellenlängen jeweils nur eine der beiden Farben vom Auge fokussiert werden. Für Rot ist das Auge weitsichtig, d.h. der Brennpunkt liegt zunächst hinter der Netzhaut und es ist eine zusätzliche Akkomodation nötig, die beim Betrachter die Empfindung von Nähe auslöst. Die Wahrnehmung von reinem Blau führt dementsprechend zur Empfindung von Ferne. Dieser Effekt ist als "chromatische Aberration" bekannt. Derartige Farbkontraste sind zu vermeiden, da sie vom Betrachter als unangenehm empfunden werden.
- Die Dichte der blauabsorbierenden Rezeptoren ist in der Fovea centralis, also dem Ort der größten Sehschärfe, vermindert, in den peripheren Sehbereichen jedoch erhöht. Die Darstellung von detailierten Formen (z.B. auch Text) in blauer Farbe ist daher zu vermeiden. Blau eignet sich aus demselben Grund jedoch besonders gut als Hintergrundfarbe (Murch 1984).
- Schwache Helligkeits- und Farbkontraste bewirken eine schlechte Lesbarkeit. Daher ist z.B. die Darstellung von Blau auf Schwarz oder Gelb auf Weiß zu vermeiden.
- Die Farbrezeptoren werden farblich umgestimmt, wenn derselbe Farbreiz längere Zeit einwirkt (chromatic adaptation). Dieser Effekt ist umso stärker, je größer der Sättigungswert des einwirkenden Farbreizes ist und je länger der Farbreiz auf die Rezeptoren einwirkt. Auf einen neuen Farbreiz reagiert ein derart umgestimmtes Farbsehsystem mit einer stark veränderten Wahrnehmung gegenüber einem normal gestimmten Farbsehsystem (Richter 1979). Es hat sich gezeigt, daß ein Benutzer die simultane Darstellung von vielen verschiedenen stark gesättigten Farben am Bildschirm bei längerer Nutzungsdauer als anstrengend empfindet.
- Die simultane Farbwahrnehmung von Infeld und Umfeld unterscheidet sich oft deutlich von der Wahrnehmung isolierter Farbreize. Umschließt z.B. eine leuchtende grüne Fläche ein graues Infeld, so erscheint dieses Infeld leicht rötlich getönt. D.h. der Farbort des farbneutralen Infelds verschiebt sich scheinbar in Richtung der Gegenfarbe des grünen Umfelds (chromatic induction). Dieser Wirkung des Simultankontrastes liegen bisher ungeklärte Ursachen von Signalveränderungen auf den ersten beiden Stufen der Farbwahrnehmung zugrunde (Ware 1982).

Die psychologische Wirkung von Farbe auf den Menschen ist in Tabelle 4.1 zusammengefaßt (Nelson 1981, Henninger et al. 1980). Es kann z.B. zum Ausgleich einer betriebsamem (=> orange) Atmosphäre die Wirkung von eisblauen Farbtönen genutzt werden.

Farbe	psychische Stimmung	Temperatur/ Distanzwirkung	Status, Attribute
Rot	anregend	warm/ nah	Gefahr, Hitze, defekt
Grün	beruhigend	kalt/ entfernt	normal, störungsfrei
Blau	passiv	kalt/ entfernt	kalt, flüssig
Gelb	warnend	warm/ entfernt	Achtung Gefahr
Orange	betriebsam	warm/ nah	

Tabelle 4.1: Psychologische Wirkung von Farbe.

Die Berücksichtigung der aufgeführten Effekte ist für die Gestaltung der Benutzeroberfläche sehr wichtig; da es sich um qualitative Prinzipien handelt, ist ein Prototyping-Ansatz - verbunden mit heuristischer Bewertung - besonders vielversprechend.

Alle genannten Phänomene der Farbwahrnehmung stehen in Bezug zu den in Bild 4.19 dargestellten Stufen der Farbwahrnehmung und lassen sich qualitativ auf einer dieser Ebenen beschreiben. Aus diesem Grund wurde zur Unterstützung der Farbgestaltung der Ansatz gewählt, in einem Werkzeug drei verschiedene Farbräume zu integrieren, die jeweils in Beziehung zu einer der genannten Stufen der Wahrnehmung stehen. Dadurch bietet dieser Ansatz die Möglichkeit, jede Stufe der Farbwahrnehmung mit einem geeigneten Farbraum zu berücksichtigen und derart unterstützt die Aufgaben der Farbauswahl effektiver bearbeiten zu können.

4.4.2 Metrische Farbräume

Die Farbmetrik befaßt sich mit der Einordnung von Farben innerhalb des Kontinuums der sichtbaren Farben mit Hilfe von Farbräumen. Man unterscheidet Farbräume der "niederen Farbmetrik", in denen Farben unabhängig vom farblichen Auflösungsvermögen des menschlichen Auges angeordnet werden. Beispiele für diese Farbräume sind der weiter unten beschriebene RGB- und HSV-Farbraum. In Farbräumen der "höheren Farbmetrik" können auch Farbabstände definiert werden; Beispiele hierfür sind die Farbräume CIELAB und CIELUV. Nachfolgend werden Farbräume vorgestellt, die in einem direkten Bezug zu den in Bild 4.19 dargestellten drei Stufen der Farbwahrnehmung stehen.

4.4.2.1 Der RGB-Farbraum

Der sogenannte RGB-Farbraum beschreibt die additive Farbmischung von Rot, Grün und Blau, indem der Anteil dieser Farbkomponenten in kartesischen Koordinaten angegeben wird. Der RGB-Farbraum steht in Beziehung zur Perzeption durch die Farbrezeptoren, d.h. zur ersten Stufe der Farbwahrnehmung (Bild 4.19). Die Menge der auf dem Bildschirm darstellbaren Farben kann innerhalb eines Würfels veranschaulicht werden (Bild 4.20).

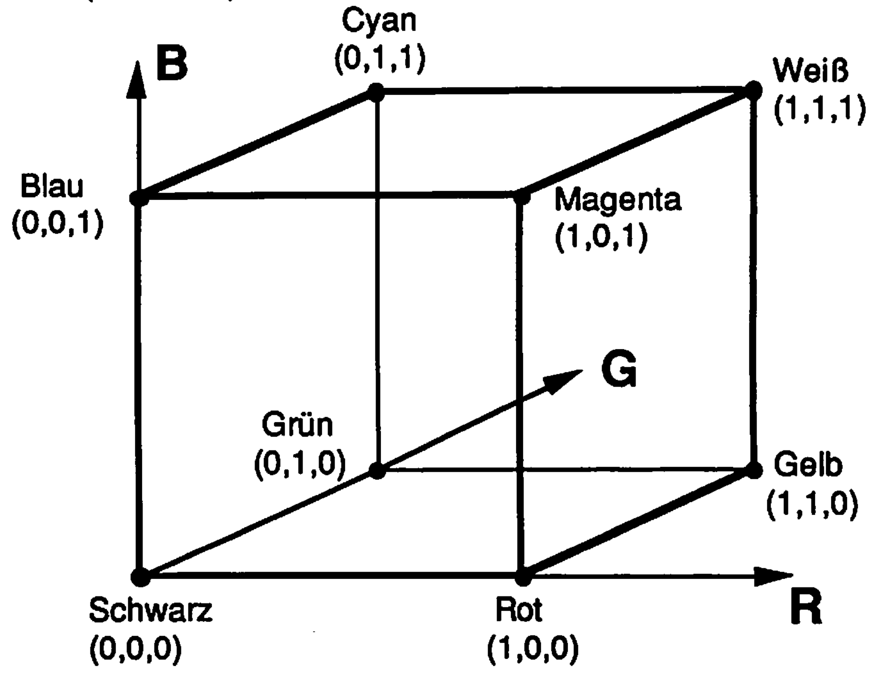


Bild 4.20: Der RGB-Farbwürfel.

Dieser Farbraum entspricht auch der bei Farbmonitoren verwendeten Technik additiver Farbmischung. Das Prinzip der additiven Farbmischung wird durch Anregung von drei verschiedenen Phosphoren in der Bildschirmmaske für jeweils ein Bildschirmpixel realisiert. Diese Phosphore werden durch drei unabhängige Elektronenstrahlkanonen angeregt.

Der Farbraum eines Monitors kann dadurch veranschaulicht werden, daß auf jeder Achse die Ansteuerungsspannung der jeweiligen Elektronenstrahlkanone aufgetragen wird. Mit einer digitalen Ansteuerung von 8 Bit Auflösung pro Elektronenstrahlkanone können 16.7 Millionen diskrete Punkte innerhalb des RGB-Würfels angesteuert werden. Der Mensch kann im gesamten Spektrum sichtbarer Farben etwa 7 Millionen Farben unterscheiden (Schmidt 1976); die Auflösung des Auges innerhalb des RGB-Farbraumes ist jedoch so unterschiedlich, daß es innerhalb des Farbraums Bereiche gibt, in denen die Farbauflösung des Monitors trotz der Zahl von 16.7 Millionen diskreten Punkten der Farbauflösung des menschlichen Auges unterlegen ist.

4.4.2.2 Der HSV-Farbraum

Der HSV-Farbraum bezieht sich auf die intuitive Farbwahrnehmung, d.h. auf die dritte in Bild 4.19 dargestellte Wahrnehmungsstufe. Im HSV-Farbraum (H=Hue, S=Saturation, V=Value) werden Farbton, Sättigung und Helligkeit einer Farbe in Zylinderkoordinaten beschrieben (Foley 1982, Bild 4.21). Der Farbton wird als Winkel zwischen 0 und 360 Grad interpretiert, Sättigung und Helligkeit liegen im Wertebereich (0..1) und werden als Radius und z-Wert dargestellt.

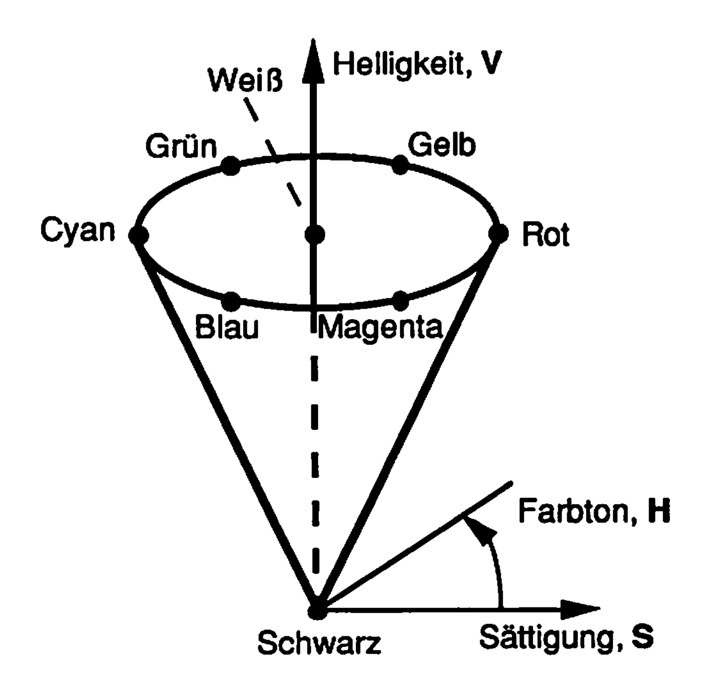


Bild 4.21: Der HSV Farbraum.

$$\boxed{H = 60 * h}$$

$$h = \begin{cases} bc - gc, & \text{if } R = MAX \\ rc - bc + 2, & \text{if } G = MAX \\ gc - rc + 4, & \text{if } B = MAX \end{cases}$$

$$rc = \frac{(MAX-R)}{d};\quad gc = \frac{(MAX-G)}{d};\quad bc = \frac{(MAX-B)}{d}$$

$$MAX = \max\{R, G, B\};\quad MIN = \min\{R, G, B\}$$

$$d = MAX - MIN$$

$$\boxed{S = \frac{d}{MAX}}$$

$$\boxed{V = MAX} \qquad (4.8)$$

Die Werte von HSV-Koordinaten können durch lineare Operationen aus den RGB-Koordinaten berechnet werden, d.h. zu jeder Kombination von HSV-Koordinaten aus dem Wertebereich (0..360, 0..1, 0..1) gehört auch genau ein RGB-Koordinatentupel aus dem Wertebereich (0..1, 0..1, 0..1). Die Umrechnung von RGB in HSV Koordinaten leisten die Gleichungen in (4.8).

Der HSV Farbraum gehört zur niederen Farbmetrik und kann daher z.B. die empfindungsgemäße Wahrnehmung von Helligkeit nicht richtig behandeln. Der Helligkeitswert V kann für jeden Farbtonwinkel den maximalen Wert 1 erreichen, jedoch wird die maximale Helligkeit eines gelben Farbtones heller empfunden als die eines blauen Farbtones; beide erhalten aber den Wert V=1.

4.4.2.3 Genormte Farbräume

Genormte Farbenräume eröffnen die Möglichkeit, eine auf dem Monitor dargestellte Farbe eindeutig zu beschreiben und damit auf anderen Monitoren reproduzierbar zu machen. Dadurch können Abhängigigkeiten von der fertigungsbedingten Streuung der Farborte der Bildschirmphosphore und der individuellen Röhrenkennlinie, die den Zusammenhang zwischen eingestellter Ansteuerungsspannung und den erzeugten Lichtquanten beschreibt, berücksichtigt werden.

Die "Commision International de l'Eclairage (CIE)" ist die international anerkannte Organisation für Normungen im Bereich von Beleuchtungsfragen. Zwei der verabschiedeten Normen werden nachfolgend vorgestellt. Das 1931 in einer CIE-Norm festgeschriebene Normvalenzsystem (DIN 5033 Teil 2) ist das Ergebnis einer langen Reihe von Untersuchungen der additiven Farbmischung dreier Primärfarbreize. Die Norm bezieht sich auf den sogenannten "farbmetrischen Normalbeobachter" der bei helladaptierten Auge zwei strukturlose Farbfelder in einem Beobachtungsfeld von 2 Grad Sehwinkel vergleicht.

Grundlage des CIE Farbraums sind drei Normfarbvalenzen (X, Y, Z), die so gewählt sind, daß alle sichtbaren Farben durch positive Zahlentripel beschrieben werden können. Zusätzlich erreicht man durch die geeignete Wahl der Normvalenzen, daß der Y-Wert ein Maß für die Helligkeit ist. Zur Kennzeichnung einer Farbe werden die Normfarbwertanteile x,y und z eingeführt:

$$x = X/(X+Y+Z) \quad y = Y/(X+Y+Z) \quad z = Z/(X+Y+Z)\,, \text{ so daß } x+y+z = 1 \qquad (4.9)$$

Mit dem Wertetupel (x, y) ist der Farbort eindeutig bestimmt, die dazugehörigen Farben unterscheiden sich dann nur noch in der Helligkeit Y.

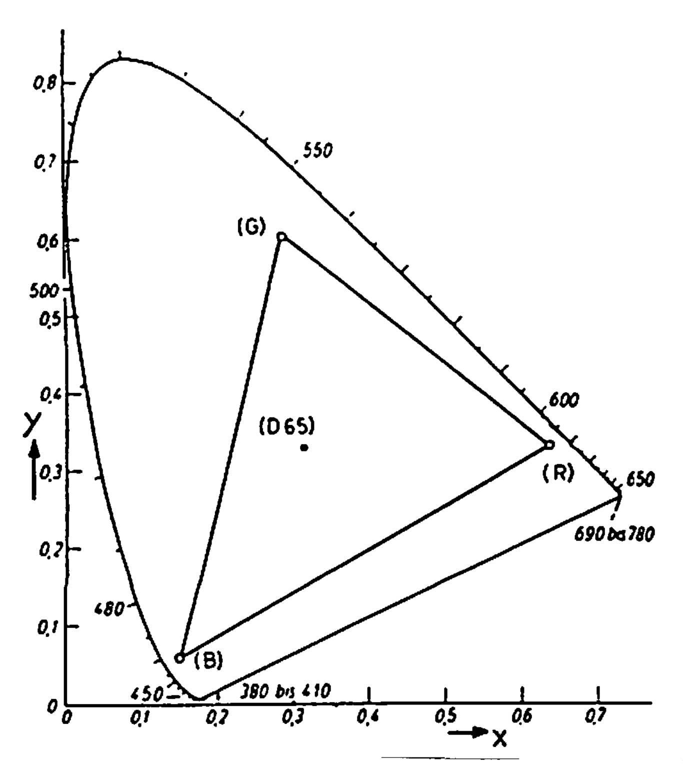


Bild 4.22: Normfarbtafel mit Monitor-Farbdreieck (aus Lang 1978).

Auf einem Farbmonitor ist nur die als Farbdreieck in Bild 4.22 eingezeichnete Untermenge aller sichtbaren Farben darstellbar, da der Farbraum durch die Koordinaten der verwendeten Bildschirmphosphore begrenzt ist. Zur Umrechnung zwischen Normfarbwerten und Bildschirmprimärvalenzen werden Transformationensmatrizen verwendet. Unter der Voraussetzung, daß die Bildschirmphosphore der EBU-Norm (European Broadcast Union) entsprechen, gelten folgende Zuammenhänge:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.4303 & 0.3416 & 0.1782 \\ 0.2219 & 0.7068 & 0.0713 \\ 0.0202 & 0.1296 & 0.9387 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \qquad (4.10)$$

Die Normfarbtafel hat die Eigenschaft, daß die additive Mischfarbe zweier Farbvalenzen immer auf deren Verbindungsgeraden liegt. Gleiche geometrische Abstände von Farborten bewirken jedoch unterschiedliche empfindungsgemäße Abstände. Ein Farbenraum mit empfindungsgemäß gleichabständigen Einheiten wurde in den Normen CIELUV und CIELAB (CIE 1978, DIN 5033 Teil 3) angestrebt. Obwohl ursprünglich CIELUV für Farbquellen und CIELAB für Körperfarben vorgesehen waren, werden heute beide Farbräume als nahezu gleichwertig angesehen. Die nachfolgenden Betrachtungen beschränken sich auf den CIELAB-Farbraum, da dieser zusätzlich einen direkten Bezug zu der Gegenfarbentheorie hat, d.h. zur zweiten Wahrnehmungsstufe in Bild 4.19. Die Koordinaten des CIELAB-Farbraums werden wie folgt bestimmt:

$$L^* = 116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16$$
$$a^* = 500\left(\left(\frac{X}{X_n}\right)^{\frac{1}{3}} - \left(\frac{Y}{Y_n}\right)^{\frac{1}{3}}\right)$$
$$b^* = 200\left(\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - \left(\frac{Z}{Z_n}\right)^{\frac{1}{3}}\right)$$
$$\text{unter der Bedingung: } \left(\frac{X}{X_n}\right), \left(\frac{Y}{Y_n}\right), \left(\frac{Z}{Z_n}\right) > 0.008856.$$
$$\Delta E = \sqrt{(L^*)^2 + (a^*)^2 + (b^*)^2} \qquad (4.11)$$

Die Normfarbwerte werden in der Gleichung jeweils auf einen Weißpunkt (Xn, Yn, Zn) bezogen. Zusätzlich sind die Begriffe Buntheit und Bunttonwinkel definiert:

$$\text{Buntheit: } \sqrt{(a^*)^2 + (b^*)^2}$$
$$\text{Bunttonwinkel: } \arctan\left(\frac{b^*}{a^*}\right) \qquad (4.12)$$

Der in Bild 4.23 dargestellte CIELAB ist in allen drei Dimensionen L*, a*, b* annähernd gleichabständig. Daher kann man den geometrischen Abstand zweier Punkte im Farbraum als deren Farbabstand (mit ΔE bezeichnet) interpretieren. Die Farbabstandsformel gilt exakt nur für kleine Abstände, wobei $\Delta E = 1$ der Wahrnehmungsschwelle entspricht (jnd = just noticeable difference). Der Vergleich großer Farbdifferenzen von jeweils 2 Farborten an verschiedenen Stellen des Farbenraumes kann nur als Näherung dienen.

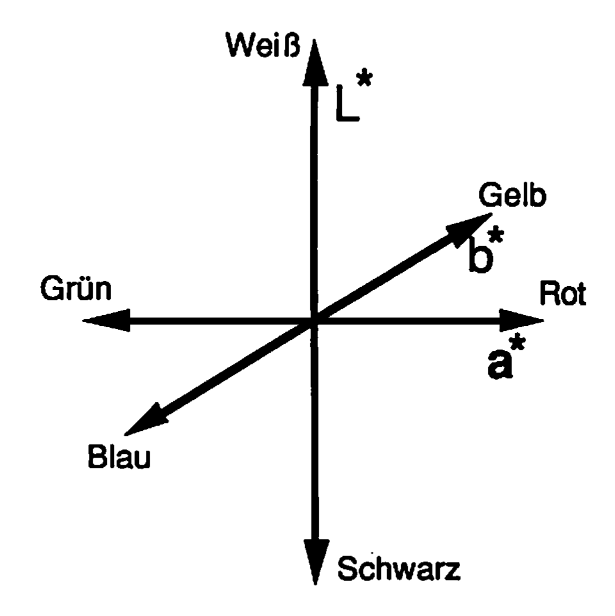


Bild 4.23: Farbenraum CIELAB.

In CIELAB repräsentieren die Koordinaten der Hauptachsen die Rot-Grün (a*), Blau-Gelb (b*) und Schwarz-Weiß (L*) Anteile einer Farbvalenz. Die dazugehörige Farbabstandsformel eröffnet generell die Möglichkeit einer quantitativen Beschreibung.

Ein anderer genormter Farbraum ist der Farbraum nach DIN 6164; er ist in 24 gleichabständige Farbstufen mit bis zu 15 Sättigungsstufen eingeteilt. Er besitzt eine eindeutige Zuordnung zu den x,y,Y Koordinaten des CIE 1931 Farbraums, die in Tabellen festgehalten ist. Ebenso sind das in den USA verbreitete Munsell-System (Newhall 1943) und das in Schweden genormte NCS-Modell (natural color system) (Hard 1969) in Tabellenform festgehalten. Diese Farbräume werden hier nicht näher besprochen, da auf Computer-Systemen die Farbraumrepräsentation durch ein geschlossenes Gleichungssystem der Tabellenform vorzuziehen ist.

4.4.2.4 Kalibrierungsverfahren für Farbmonitore

Die Verwendung eines genormten Farbenraums erfordert die Bestimmung von gerätespezifischen Parametern eines Farbmonitors.

a) Bestimmung der Bildschirmphoshore

Da bisweilen fertigungsbedingte Abweichungen von den vom Hersteller angegebenen Farborten auftreten, wurden diese Angaben meßtechnisch mit Hilfe eines Photometers überprüft. Die von der European Broadcast Union (EBU) als Norm vorgegebenen Werte werden, wie Tabelle 4.2 zeigt, nicht exakt von den Monitorherstellen eingehalten, so daß die Matrix in Gleichung 4.10 nicht exakt stimmt.

Farbe	EBU-Norm		HP-19' Monitor		HP-16' Monitor	
	x	y	x	y	x	y
Rot	0.640	0.330	0.627	0.332	0.615	0.341
Grün	0.290	0.600	0.315	0.603	0.289	0.598
Blau	0.150	0.060	0.148	0.059	0.150	0.073
Weiß	0.313	0.329	0.299	0.295	0.294	0.292
Gamma			2.9		2.3	

Tabelle 4.2: Farborte nach EBU-Norm und gemessene Werte.

Die Abweichungen der Phosphorfarborte haben insgesamt jedoch geringere Auswirkungen als die Unterschiede der Röhrenkennlinien.

b) Bestimmung der Röhrenkennlinie
Der Zusammenhang zwischen Elektronenstrahlstrom und Strahlungsfluß der erzeugten Lichtquanten ist streng linear, jedoch ist der Zusammenhang zwischen angelegter Spannung und Elektronenstrom exponentiell. Die Übertragungsfunktionen sind in Bild 4.24 dargestellt.

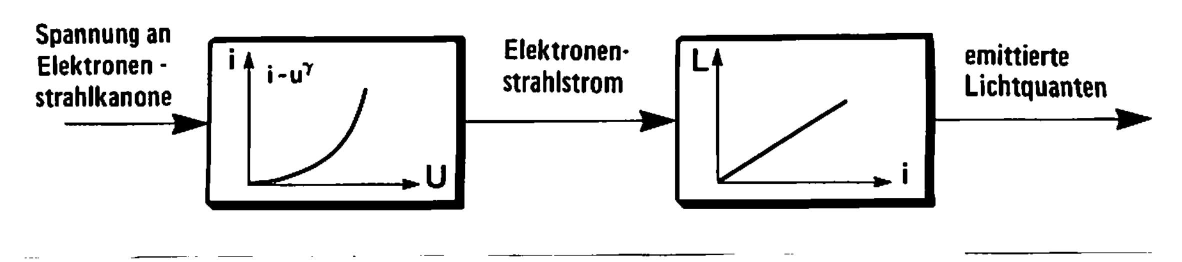


Bild 4.24: Zusammenhang zwischen Spannung und Lichtquanten.

Alle rechnerisch ermittelten RGB-Werte müssen also mit 1/gamma potenziert werden, um die Nichtlinearität der Bildröhre auszugleichen. Dies entspricht der Vorverzerrung, die bei der Übertragung eines Fernsehbildes durch die Aufnahmekamera geleistet wird. Zur Bestimmung des Gamma-Wertes müssen die Farborte der Phosphore bereits bekannt sein, so daß die Transformationsmatrix in Gleichung 4.10 festgelegt ist. Durch iterativen Vergleich zwischen dem aus RGB-Werten der Ansteuerung errechneten Farbort unter Annahme eines gamma-Wertes und dem tatsächlich gemessenen Farbort kann der zunächst geschätzte Gamma-Wert bestimmt werden.

Abschließend sind die wichtigsten Eigenschaften der verschiedenen Farbräume in Tabelle 4.3 zusammengefaßt:

	RGB	HSV	CIE	CIELAB	DIN 6164
rechnergerechtes Gleichungssystem	•	•	•	•	
Rezeptor-bezogen	•				
Gegenfarben-bezogen				•	
intuitiv		•			•
gleichabständig				•	•
genormt			•	•	•

Tabelle 4.3: Übersicht wichtiger Eigenschaften der Farbräume.

4.4.3 Konzept des Farbeditors CEDI

In diesem Abschnitt wird das Konzept für einen interaktiven Farbeditor aufgestellt, der den iterativen Prozeß der Farbgestaltung unterstützen soll. Aufbauend auf den letzten beiden Abschnitten werden für dieses Werkzeug folgende Anforderungen formuliert:

- interaktive Auswahl und quantitative Bestimmung isolierter Farbreize mit Hilfe geeigneter Farbräume (RGB, HSV, CIELAB),
- Speicherung von Farbwerten in einer Farbdatenbank,
- Berücksichtigung des Simultankontrastes für verschiedene Formen und Größen,
- Einsatzmöglichkeit unabhängig von der Applikation,
- Portierbarkeit von Farbwerten mit Hilfe eines normierten Farbraums.

Bevor die Merkmale des Werkzeugs CEDI beschrieben werden, wird eine kurze Charakterisierung von anderen, bereits eingesetzten Werkzeugen vorgenommen.

4.4.3.1 Werkzeuge zur Farbgestaltung

Farbtüchtige Computer wie etwa der Apple-Macintosh II verfügen mit einer entsprechenden Graphikkarte bereits über einen Farbumfang von 16.7 Millionen Farben. Der Apple-Mac II bietet ein Werkzeug zur interaktiven Farbauswahl; es nutzt den HSV-Farbraum zur Visualisierung des Farbangebots mittels eines "Farbrades" und einem "Rollbalken" für die Helligkeitswahl. Eine Normierung der Farbwerte oder eine Farb-Datenbank sind nicht vorgesehen. Die Farbauswahl bezieht sich immer nur auf eine isolierte Farbe, die dann z.B. für die Hervorhebung aktivierter Elemente oder den Bildschirmhintergrund vergeben wird. Der Einsatz dieses Werkzeug innerhalb eines Anwenderprogramms kann nur durch umfangreiche Programmierarbeiten erreicht werden.

Kaster et al. (1985) entwickelten ein Werkzeug zur interaktiven Manipulation von Bildschirmdarstellungen, das auf einem zweiten Bildschirm separat bedient wird. Dort wird mit einem genormten Farbenraum, dem gleichabständigen CIELUV gearbeitet. Das Werkzeug wurde speziell für die farbliche Gestaltung von computergespeicherten Landkarten entwickelt und beschränkt sich daher auf die Farbmanipulation von 6 kartographischen Elementen (Wasser, Stadt, etc.). Dieses Werkzeug ist eng auf die kartographische Farbgestaltung zugeschnitten und daher für andere Applikationen in dieser Form nicht einsetzbar.

Die genannten Werkzeuge können nicht die aufgestellten Anforderungen, die für ein iteratives Vorgehen bei der Farbgestaltung notwendig sind, erfüllen. Daher wurde ein neues Werkzeug entwickelt, das alle genannten Anforderungen berücksichtigt.

4.4.3.2 Beschreibung des Werkzeugs CEDI

Das Werkzeug CEDI (Color EDItor) ist so entworfen, daß eine interaktive Manipulation von Farbe in den 3 vorgestellten Farbräumen RGB, HSV und CIELAB möglich ist. Das Ergebnis dieser Manipulation wird kontinuierlich in entsprechenden Farbflächen auf dem Monitor visualisiert.

Die Koordinaten einer Farbe können festgelegt und in jedem der drei Farbenräume berechnet und angezeigt werden. Im Gegensatz zu anderen Werkzeugen wird nicht nur ein isolierter Farbreiz bearbeitet, sondern es können zunächst zwei Farbwerte unabhängig voneinander bearbeitet und sowohl einzeln als auch simultan dargestellt werden. Die Simultandarstellung ist in 3 unterschiedlichen Formen möglich, die jeweils eine für ein Bildschirmlayout typische Kontrastsituation wiedergeben.

1. Rechteck innerhalb eines Rechtecks (Infeld/Umfeld),
2. Linienzug innerhalb eines Rechtecks und
3. Textdarstellung innerhalb eines Rechtecks.

Die Größe des Infeldrechtecks, die Liniendicke und die gewählte Schriftart (Textfont) können interaktiv verändert werden. Damit kann kontrolliert und unabhängig voneinander die subjektive Wirkung des simultanen Farbkontrastes in Verbindung mit unterschiedlichen Formen und Größen interaktiv variiert und beurteilt werden.

Ein weiteres Merkmal von CEDI ist die Ankopplung einer Datenbank, in der Farbkoordinaten unter einem Bezeichner abgespeichert oder gelesen werden können. Die Struktur dieser Farbdatenbank ist der im Konzept von X-Windows (Gettys et. al. 1988) vorgesehenen Datei "/usr/lib/rgb.txt" ähnlich; dort sind Name eines Farbtons und die entsprechenden RGB-Werte in einer Textdatei abgespeichert. Jedoch wird dort die individuelle Verzerrung durch die Bildröhrenkennlinie nicht berücksichtigt, so daß eine Vereinheitlichung der Farbgestaltung für jedes System tatsächlich nicht gegeben ist. Die Anpassung der RGB-Werte dieser Datei wird demjenigen überlassen, der den "X-Server" auf einem Computersystem implementiert (Nye 1988). Im Gegensatz dazu speichert CEDI vom Gamma-Faktor unabhängige RGB-Werte, so daß die Übertragung von Farbwerten zwischen Monitoren mit ähnlichen Bildschirmphosphoren (wie in Tabelle 4.2) gewährleistet ist. Bei einer großen Streuung der Farbwerte der Phosphore kann die Speicherung der CIELAB Koordinaten vorgenommen werden. Die in der Farbdatenbank von CEDI benutzten Bezeichner können Farbbezeichner ("skyblue") sein, wie sie auch in der X-Windows Farbdatenbank vorgesehen sind. Jedoch können auch Bezeichner gewählt werden, die direkt den Kontext ausdrücken, in dem eine Farbkodierung verwendet werden soll (z.B. "cooling water", "infusion-pipe").

4.4.3.3 Interaktion mit CEDI

Die Benutzung des Farbeditors CEDI erfolgt direkt-manipulativ durch Mauseingabe mittels virtueller Tasten und Werteschieber auf dem Farbbildschirm. Das Werkzeug CEDI verfügt auf der linken Seite über eine Menüleiste, aus der einer der 3 Farbräume (RGB, HSV oder CIELAB) und eine der 3 Kontrastsituationen ausgewählt werden. In Bild 4.25 wurde der HSV-Farbraum zur interaktiven Manipulation gewählt; als Simultandarstellung im mittleren Farbfenster (A) ist eine Infeld/Umfeld Kontrastsituation dargestellt. Die Farbwerte des linken (B) oder rechten Farbrechtecks (C) können mit Hilfe der Maus an den jeweiligen Werteschiebern physikalisch unabhängig verändert und ihr Zusammenwirken im mittleren Rechteck (A) beurteilt werden.

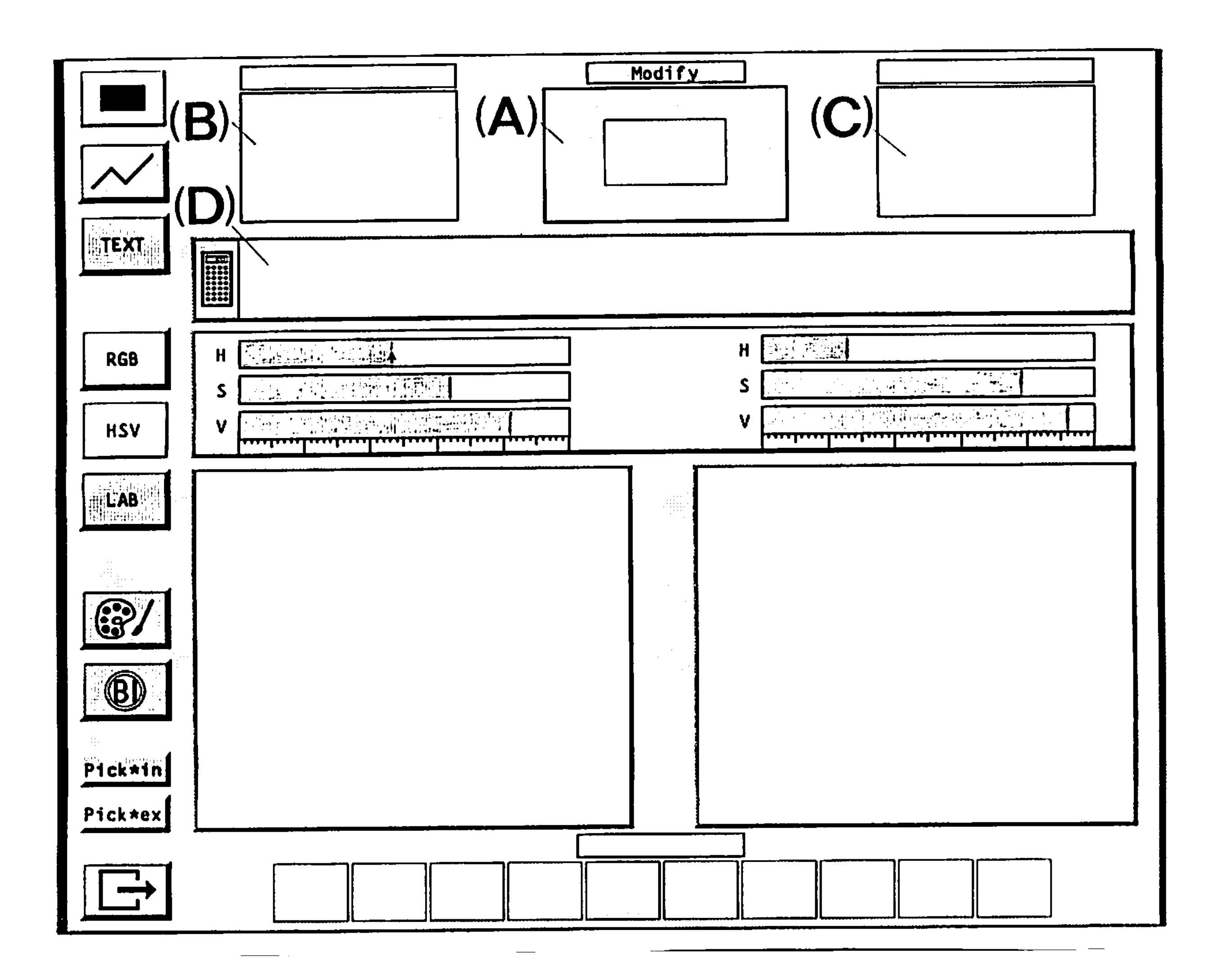


Bild 4.25: Bildschirmlayout des Werkzeugs "CEDI".

Oberhalb der beiden Farbfelder (B) und (C) befindet sich ein Texteingabefenster für Zugriffe auf die Farbdatenbank. Wenn dieses Texteingabefenster per Maus aktiviert wurde, kann ein Farbbezeichner über die Tastatur eingegeben werden. Durch Aktivieren eines Displays (D) oberhalb der Werteschieber kann die numerische Bestimmung der eingestellten Farbwerte durch Koordinatentripel der 3 Farbenräume angezeigt werden (Farbbild 4.2, Anhang B-3). In der Mitte des Fensters wird der in CIELAB berechnete Farbabstand der beiden Farben angegeben. Dieser Farbabstand soll gemäß ISO 9241 mindestens $\Delta E = 40$ betragen, um einen ausreichenden Farbkontrast zweier Elemente zu erreichen. In Farbbild 4.2 sind die eingestellten Farbwerte im CIELAB-Farbraum visualisiert. Der Helligkeitswert ist auf einem vertikalen Werteschieber aufgetragen (L-Koordinate), der Buntheitswert (a- und b-Koordinaten) ist innerhalb der grauen Fläche markiert. Diese Fläche ist ein horizontaler Schnitt durch den Farbenkörper in Höhe des aktuellen Helligkeitswertes (vgl. Bild 4.23). Diese Darstellung wird auch zur interaktiven Manipulation einer Farbe im CIELAB-Farbraum benutzt.

Am unteren Rand des Bildes erscheint ein gleichmäßig gestufter Farbübergang zwischen den beiden Farben (lineare Abstufung in CIELAB). Jede dieser dargestellten Farben kann interaktiv mit der Funktionstaste "Pick*in" als neue Anfangs- oder Endfarbe ausgewählt werden, so daß derart modifiziert ein neuer Farbübergang festgelegt und quantifiziert werden kann. Diese Interaktionssequenz erleichert das Auffinden einer Farbnuance, von der man nur weiß, daß sie zwischen zwei bekannten Farbtönen liegen soll. Die dargestellten Farbwerte des Farbübergangs können auch als Einheit ebenso wie einzelne Farbwerte in der Farbdatenbank unter einem Bezeichner abgespeichert oder geladen werden. Derart bestimmte Farbübergänge können z.B. dazu benutzt werden, kontinuierliche Änderungen in Prozeßzuständen zu visualisieren.

4.4.3.4 Direkte Rückkopplung der interaktiven Farbmanipulation

Die interaktive Manipulation einer Farbdarstellung mittels CEDI soll mit einer direkten Rückkopplung dieser Manipulation einhergehen. Hierzu wird die "indirekte" Farbzuweisung mittels einer Farbtabelle genutzt. Es wird für ein Darstellungselement lediglich die Adresse in einer Farbtabelle festgelegt, auf der ein RGB-Tripel zu finden ist. Viele moderne Farbgrafiksysteme adressieren 256 Einträge in der Farbtabelle, die jeweils mit 3 Byte Farbinformation belegt sind (Bild 4.27).

Diese indirekte Farbzuweisung hat den großen Vorteil, daß durch Neubelegung der Einträge in der Farbtabelle Änderungen direkt ohne Neuzeichnen sichtbar werden; jedes zuvor mit dieser Adresse gezeichnete Objekt wird gleichzeitig verändert. Im Bildschirmentwurf können dadurch zusammenhängende Flächen oder gleichartige

Objekte, die mit derselben Farbadresse gezeichnet wurden, auch simultan farblich variiert werden.

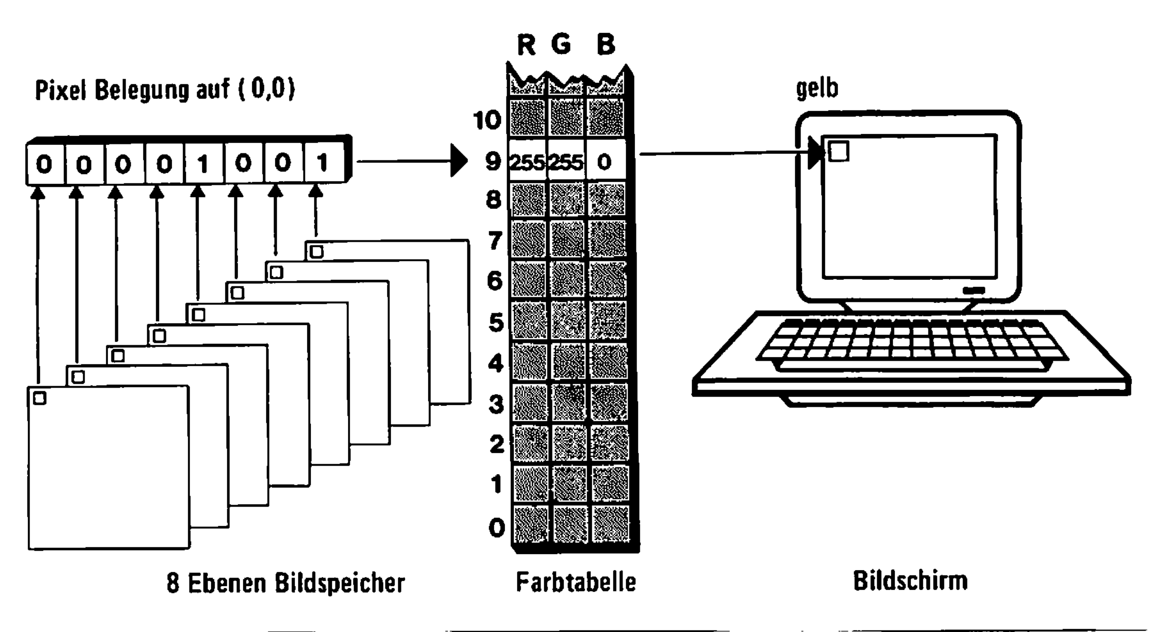


Bild 4.27: Farbwertzuweisung mittels einer Farbtabelle.

4.4.4 Anwendung des Farbeditors CEDI

4.4.4.1 Beispiele für interaktive Farbgestaltung

Nachfolgend werden zwei Beispiele vorgestellt, die die Möglichkeiten von CEDI anhand einer konkreten Problemstellung erläutern.

- Die Zustände der Interaktionselemente ("aktiv", "aktivierbar" und "nicht aktivierbar") sind zur Benutzerführung farblich kodiert. Als Farbton wird für alle drei Zustände Blau gewählt, da Blautöne als Hintergrundfarben - wie bereits erwähnt wurde - besonders geeignet sind. Das optische Hervortreten eines Interaktionselements kann durch starke Helligkeit und/oder Sättigung eines Farbtons erreicht werden. Der Zustand "aktiv" sollte daher die größte Helligkeit erhalten; der Zustand "aktivierbar" sollte einen helleren und gesättigteren Farbton erhalten als der Zustand "nicht aktivierbar". Die Eigenschaften der Farbräume HSV und CIELAB ermöglichten es, den Farbton dieser Zustände gleich zu wählen und die Sättigungs- und Helligkeitsdifferenz geeignet abzustufen. Dadurch bietet die Benutzeroberfläche ein farblich konsistentes Erscheinungsbild. Die ermittelten Farbwerte für die drei Aktivierungszustände wurden unter den Bezeichnern "active", "standby" und "locked" in die Datenbank eingetragen.
- Bei der Gestaltung der Benutzeroberfläche eines Anästhesie-Entscheidungsunterstützungs Systems (AES-2) soll die Möglichkeit geschaffen werden, sog. "intelli-

gente Alarme" zu visualisieren. Dazu wurde ein kontinuierlicher Farbübergang von einem grünen Farbton nach Rot bestimmt, so daß eine zunehmende Abweichung vom Normalzustand durch den wachsenden Rotanteil in einer speziellen Balkengrafik angezeigt wird. Die Helligkeit des Farbreizes wird nach Rot hin geringer, um auch rotgrünblinden Benutzern mindestens eine zusätzliche Kodierungsdimension zur Verfügung zu stellen.

Die geschilderten Gestaltungsaufgaben sind durch heuristisches Vorgehen ohne analytische Werkzeuge nur mit großem Zeitaufwand lösbar.

4.4.4.2 Farbliche Feinabstimmung einer Applikation

Die Erfahrung beim praktischen Arbeiten mit CEDI hat gezeigt, daß sich zunächst die Farbauswahl für bestimmte Infeld/Umfeld Konstellationen effektiv lösen läßt. Später wird dann eine Feinabstimmung der Farben in bezug auf das gesamte Bildschirmlayout der Applikation erforderlich, da die Effekte einer simultanen Darstellung nicht analytisch erfaßbar und damit in ihrer Wirkung auf den Benutzer nicht vorhersagbar sind. Aus diesem Grund wurden für die farbliche Feinabstimmung eines gesamten Bildschirms zwei Methoden entwickelt: eine 1-Bildschirm und eine 2-Bildschirm Lösung.

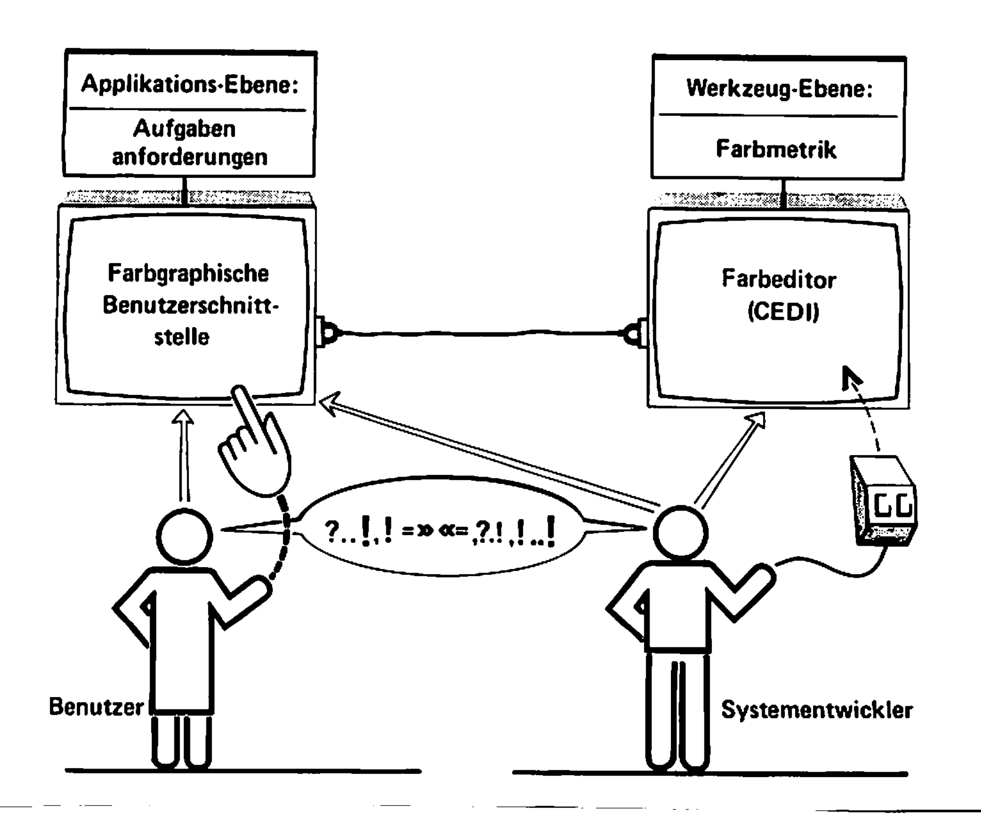


Bild 4.28: Konfiguration mit zwei Bildschirmen.

Die Konfiguration mit zwei Bildschirmen sieht vor, CEDI separat auf einem zweiten Bildschirm neben dem Bildschirm der Applikation zu starten (Bild 4.28). Die Bildschirme sind über ein Netzwerk miteinander verbunden, so daß ein interaktiv ausgewählter Farbwert eines Bildschirmpunktes der Applikation in das Werkzeug CEDI übertragen werden kann (Funktionstaste "Pick*ex"). Die Modifikation wird dann wie gewohnt in einem der drei Farbräume durchgeführt, wobei jede Änderung auch in der Applikation direkt sichtbar wird. Diese Konfiguration ermöglicht also einen Feinabgleich der Farben, ohne daß die Geometrie der Applikation durch das Werkzeug beeinflußt wird.

Für den Fall, daß nur ein Bildschirm zur Verfügung steht, wurde das interaktive Werkzeug CEDI-FT (FT=Fine-Tuning) entwickelt. CEDI-FT ist funktional und geometrisch eine Teilmenge von CEDI und kann einer Bildschirmdarstellung direkt überlagert werden. Die Vorgehensweise bei der Farbmanipulation erfolgt dann wie in CEDI durch die drei Werteschieber (Bild 4.29). Eine Farbmanipulation kann im RGB- oder HSV-Farbraum vorgenommen werden. Die Farbkoordinaten sowie die aktuell manipulierte Farbadresse werden in einem Textfeld eingeblendet. Die Taste "Move" ermöglicht das Verschieben von CEDI-FT auf dem Bildschirm, um unerwünschte Überlagerungen zu vermeiden.

Die Erfahrung während des Prototypings verschiedener Benutzeroberflächen zeigte, daß insbesondere die 2-Bildschirm Konfiguration für den Feinabgleich effektiv zur Diskussion der Farbgestaltung beitragen konnte. Bei der Überarbeitung des Anästhesie-Information-Systems AIS konnte der Systemdesigner gemeinsam mit potentiellen Benutzern die farbliche Gestaltung diskutieren und variieren und so eine nach Urteil aller Beteiligten bessere Farbkomposition gefunden werden (siehe Abschnitt 4.4.5).

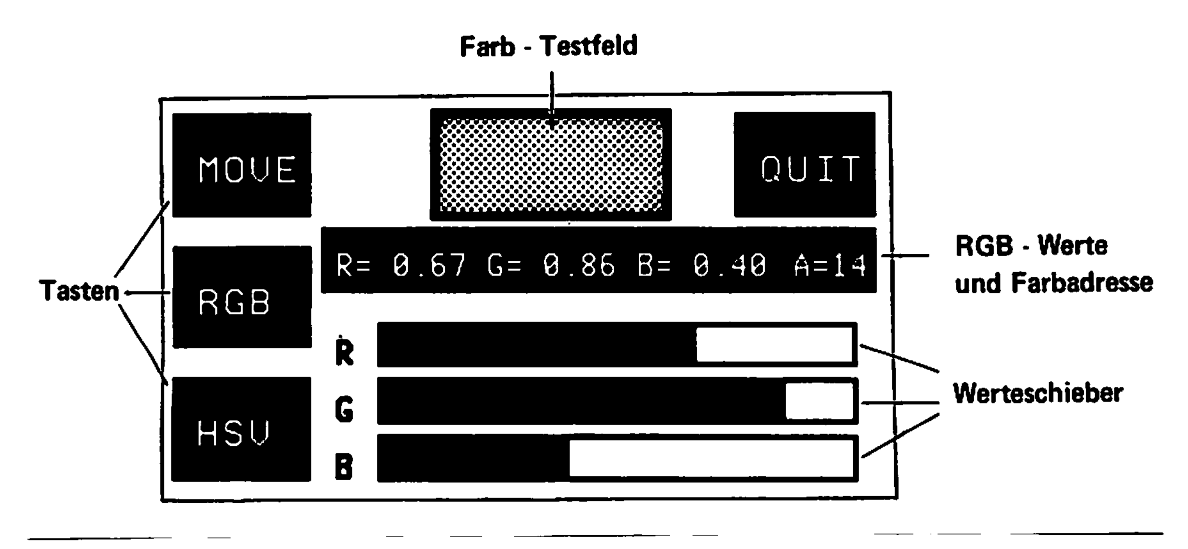


Bild 4.29: Bildschirmlayout des Werkzeugs "CEDI-FT".

4.4.5 Analytische Bewertung farbiger Bildschirmdarstellungen

Die bisher beschriebenen Werkzeugfunktionen unterstützen die subjektive Beurteilung einer Farbgestaltung. Darüber hinaus wird in dieser Arbeit untersucht, wie eine Bewertung auch auf quantitative Werte gestützt durchgeführt werden kann.

4.4.5.1 Farbenalphabet versus komplexe Graphik

Mit dem Kriterium einer "sicheren Unterscheidbarkeit" (>90 %) von Farben wurde in verschiedenen Arbeiten (Meister und Sullivan 1969, Häusing 1975) ein Farbenalphabet bestimmt, das die Farborte der hierfür optimalen Farben angibt. Bei dieser Zielsetzung wird Farbe also hinsichtlich der Kategorisierung von Information genutzt. Für Bildschirmgeräte wird ein Farbumfang (Farbenalphabeth) mit 4-6 Farborten vorgeschlagen (vgl. auch Miller's Gesetz "7±2"). Diese Ergebnisse finden Anwendung in der Gestaltung reiner Textdisplays wie z.B. für BTX-Endgeräte (van Nes 1985) oder in der Empfehlung, nicht mehr als 6 farbige Kurven innerhalb eines xy-Diagramms aufzutragen. Aufgrund dieser Ergebnisse wird fälschlicher Weise häufig angenommen, daß generell die Benutzung von mehr als 6 Farben auf dem Bildschirm ergonomisch ungünstig wäre. "Farbe" verfügt jedoch über weit mehr Eigenschaften als die der Kategorisierung. Das folgende Zitat von Treisman (1986) verdeutlicht dies:

> "In contrast to identification and classification, discrimination between simultaneously present stimuli, where memory is only minimally involved, more often reflects dimensional processing, with fine and continuous gradation of information. We can discriminate thousands of pairs of tones or flashes. Aesthetic responses to a Gauguin painting would be very different if it were reproduced in just four or five different color tones."

Bei der Darstellung von Zusammenhängen, graduellen Veränderungen oder zum Zwecke der Benutzerführung kann also durchaus eine größere Anzahl von Farben auf einem hochauflösenden Graphikdisplay sinnvoll sein. Auch die Annahme, daß eine höhere Anzahl von Farben zu schnellerer Ermüdung führt, blieb bislang unbegründet (Smith 1987).

4.4.5.2 Quantitative Größen farbiger Darstellungen

Für komplexe farbgraphische Darstellungen bietet ein Farbenalphabeth bei der Gestaltung und Bewertung keine Hilfe, so daß man bisher die in Abschnitt 4.4.1 genannten qualitativen Prinzipien für die Gestaltung und die Benutzerbefragung *für die* Bewertung einsetzt. Eine quantitative Kenngröße zur Bewertung, die aus den im Bildschirmentwurf verwendeten Farbwerten abgeleitet wird, würde dem Entwickler

jedoch bereits während des Entwurfs eine unmittelbare Rückkopplung bieten und so die Möglichkeit zur Voroptimierung geben.

Aus Diskussionen mit Benutzern war bekannt, daß schlechte Bildschirmdarstellungen häufig als "zu bunt" oder "grell" kommentiert wurden. Die Erfahrung bei der Farbgestaltung eines Anästhesie-Informations-Systems (AIS) (Klocke 1987) hat gezeigt, daß besonders die Wahl der niedrigen Sättigungswerte dafür verantwortlich ist, daß die Bildschirmgestaltung als "angenehm" bezeichnet wird. Aus diesen Gründen wurden zunächst die Sättigungswerte einer farbigen Darstellung näher betrachtet. Die im CIELAB Farbraum definierte Buntheit (Gleichung 4.12) entspricht dem Begriff der Sättigung und wird daher als Ausgangspunkt für eine Analyse gewählt. Der Mittelwert aller Pixel-Buntheiten bezogen auf einen ausgewählten Bildschirmbereich von m Zeilen und n Spalten wird nach Gleichung 4.13 bestimmt.

$$\overline{B} = \frac{1}{m^* n} \sum_{s=1}^{n} \sum_{z=1}^{m} \sqrt{a_{zs}^2 + b_{zs}^2} \qquad (4.13)$$

Die Bedeutung der Buntheit wird zunächst anhand einer Farboptimierung im Anästhesie Informations System AIS betrachtet. Im Zuge einer Portierung des AIS auf ein neues Graphiksystem sollte die farbliche Gestaltung weiter verbessert werden. Die Optimierung der Farbgebung wurde mit Hilfe des Farbeditors CEDI durchgeführt. Die Modifikation einzelner Farben wurde solange fortgesetzt bis jeder aus der daran beteiligten Gruppe von Benutzern und Entwicklern mit der Gestaltung der verschiedenen Arbeitsseiten des AIS zufrieden war. Die errechneten mittleren Buntheiten der Arbeitsseiten vor und nach der Optimierung sind in Tab 4.4 zusammengefaßt.

Zunächst fällt auf, daß die Abweichungen der Buntheitswerte auf verschiedenen Arbeitsseiten für beide Versionen relativ gering sind. Dies bedeutet, daß bei einem Seitenwechsel das Auge stets mit ähnlichen mittleren Sättigungswerten konfrontiert wird. Diese Konsistenz in der Sättigung wird von den Benutzern als sehr positiv beurteilt. Es kann daher angenommen werden, daß geringe Abweichungen der mittleren Buntheit auf eine harmonische Farbgestaltung hinweisen.

Die Daten in Tab. 4.4 zeigen weiter, daß die Überarbeitung der Farbgestaltung mit einer Reduzierung der mittleren Buntheit um durchschnittlich 22 % einherging. Damit wurde in diesem Fall die Vermutung bestärkt, daß geringere Sättigungswerte zu einer besseren Beurteilung führen und daß die Buntheit hierfür eine sinnvolle Kenngröße ist.

Seite	$\bar{B}_{vorher}$	$\bar{B}_{nachher}$
Systemseite	20.3	17.2
Hauptverteilerseite	24.3	19.8
Arbeitsseite 7	23.4	18.3
Textseite	24.7	18.5
Patientenseite	23.1	16.9
Konfigurationsseite	19.5	16.0
Mittelwert	22.6	17.8

Tab. 4.4: Buntheitswerte des AIS vor und nach der Optimierung.

4.4.5.3 Studie zur subjektiven Bewertung farbiger Darstellungen

Auf die Ergebnisse des letzten Abschnitts aufbauend wurde eine Studie geplant und durchgeführt, um den Zusammenhang zwischen der Buntheit und der subjektiven Bewertung einer farblichen Gestaltung durch den Benutzer konkreter zu erfassen.

Der Entwurf für die Benutzerschnittstelle der Infusionspumpen-Kontrolleinheit aus Kapitel 3.3 wurde hierfür von verschiedenen Personen farblich gestaltet. Insgesamt wurden 8 verschiedene Versionen erstellt; davon sind 4 von Designern(innen), 2 von Ärzten(innen) und 2 vom Entwicklerteam (Ingenieure und Informatiker) entworfen worden. Alle Personen erhielten die Anweisung, eine in ihren Augen für die Bedienung in einer Intensivstation geeignete farbliche Gestaltung vorzunehmen. Für die Gestaltung, die mit Hilfe von CEDI-FT durchgeführt wurde, gab es kein Zeitlimit. Diese Entwürfe wurden dann von medizinischem Personal (12 Personen) nach verschiedenen Kriterien bewertet, die auf einer bipolaren Skala von -3 bis +3 einzuschätzen waren:

abzuraten empfehlenswert (K1)
-3 -2 -1 0 +1 +2 +3
aufdringlich dezent (K2)
-3 -2 -1 0 +1 +2 +3
zu farbig zu blaß (K3)
-3 -2 -1 0 +1 +2 +3

Die Begriffe wurden in Anlehnung an Widdel (1987) gewählt, der die Farbwirkung nach den Faktoren Brauchbarkeit, Ausdruck und Ästhetik differenzierte. So bezieht sich das Adjektiv "empfehlenswert" auf die Brauchbarkeit und "aufdringlich" auf den Ausdruck einer Farbwirkung.

Nr.	$\bar{B}$	$B_{\sigma-1}$	K1	K2	K3
1	10.1	14.3	0.1	1.2	1.5
2	12.9	20.5	1.7	0.2	0.0
3	14.2	12.1	2.1	1.3	0.4
4	18.6	19.4	-0.5	0.8	0.5
5	21.0	11.0	0.3	0.9	0.4
6	26.3	37.4	-1.4	-2.0	-2.2
7	35.5	10.0	-0.8	-1.0	-1.7
8	76.8	24.6	-2.6	-2.7	-2.7

Tab. 4.5: Buntheiten und Bewertungen verschiedener Entwürfe.

In Tab. 4.5 sind für die verschiedenen Entwürfe die Werte der mittleren Buntheit und deren Standardabweichung und die Mittelwerte der Beurteilungen für K1, K2 und K3 angegeben. Nach den Ergebnissen des letzten Abschnitts wäre im einfachsten Fall eine abnehmende Bewertung mit zunehmender Buntheit zu erwarten. Bild Nr. 6 fügt sich aber nicht in diese Reihe ein; die Besonderheit von Nr. 6 ist eine hohe Standardabweichung der Buntheit, so daß diese als weitere Kenngröße hinzugezogen wurde. Es wurde die These aufgestellt, daß neben dem Mittelwert ebenso eine große Standardabweichung $B_{\sigma-1}$ der Buntheit zu negativen Bewertungen führt.

Die Berücksichtigung von 2 Variablen für eine abhängige Größe kann durch multiple lineare Regression beschrieben werden. Zunächst werden die Ergebnisse der Regressionsanalyse für die Kriterien K2 und K3 angegeben:

$$K2 = 3.04 - 0.069\,\bar{B} - 0.081\,B_{\sigma-1} \qquad (4.14)$$

Regressionskoeffizient: R^2=0.92 Signifikanz: $p < 0.025$

$$K3 = 2.42 - 0.067\,\bar{B} - 0.066\,B_{\sigma-1} \qquad (4.15)$$

Regressionskoeffizient: R^2=0.84 Signifikanz: $p < 0.1$

Ein derart linearer Zusammenhang kann für das Kriterium K1 (Brauchbarkeit) nicht gelten; ein graues Bild ($\bar{B} = B_{\sigma-1} = 0$) würde die höchste Bewertung erhalten. Die Bewertung von Nr.1 zeigt, daß trotz geringer mittleren Buntheit und Standardabweichung die Bewertung nur schwach positiv ist. Die verwendeten Farben liegen also schon zu nahe an den Grautönen; die hohe Bewertung in K3 ("zu blaß") bestätigt diese Vermutung. Nach Eliminieren von Nr. 1 aus der Regressionsanalyse konnte für die übrigen Werte ein signifikanter linearer Zusammenhang bestimmt werden:

$$K1 = 2.97 - 0.075\,\bar{\bar{B}} - 0.059\,B_{\sigma-1} \qquad R^2{=}0.81 \qquad p < 0.1 \qquad (4.16)$$

Die Bedeutung von Mittelwert und Standardabweichung der Buntheit ist für die verschiedenen Kriterien in etwa gleichgewichtig. Ein geringer Mittelwert von der Buntheit und deren Standardabweichung steigert zwar einerseits die Werte für K1 und K2 ("empfehlenswert" und "dezent") gleichzeitig tendiert aber der Wert von K3 zu der Beurteilung "zu blaß". Eine optimale Kombination von $\bar{\bar{B}}$ und $B_{\sigma-1}$, die zu der Bewertung "empfehlenswert und dezent, aber nicht zu blaß" führen (Bild Nr. 3), kann daher durch die heuristisch gefundenen Gleichungen nicht vorhergesagt werden. Sie bieten jedoch die Möglichkeit, nach Ermittlung der Buntheiten eines neuen Entwurfs abzuschätzen, wie ein Benutzer die Kriterien K1, K2 und K3 ungefähr bewerten würde.

Zur Überprüfung der Ergebnisse wurden die ermittelten Gleichungen auf ein weiteres Farblayout angewendet. Neben den 8 Entwürfen in Tab. 4.5 wurde auch die Arbeitsseite 7 des AIS in der Studie bewertet. Diese AIS-Arbeitsseite unterscheidet sich in Größe und Farbanzahl deutlich von der Infusionspumpen-Kontrolleinheit; es werden 21 verschiedene Farben in einem Bildschirm-füllenden Layout verwendet.

$\bar{\bar{B}}$	$B_{\sigma-1}$	K1	K2	K3	
18.3	13.7	0.8	0.7	0.3	errechnete Werte
		1.5	0.4	0.2	Bewertung aus der Studie

Tab. 4.6: Errechnete und in der Studie ermittelte Bewertung einer AIS Arbeitsseite.

Die Ergebnisse der Gleichungen (4.14) - (4.16) liefern die in Tab. 4.6 aufgeführten Werte für K1, K2 und K3. Im Vergleich mit den in der Studie ermittelten Bewertungen liegen die errechneten ("geschätzten") Werte tendenziell richtig, d.h. die Ergebnisse der Studie sind mit den genannten Einschränkungen auch auf neue Entwürfe übertragbar. Gleichzeitig wurde die These bestärkt, daß die Anzahl der Farbtöne keine negative Einflußgröße darstellen muß; die Bewertung der Qualität der AIS Arbeitsseite mit 21 Farben erhält einen Wert von 1.5 während der geschätzte Wert auf Grundlage eines Entwurfs mit nur 8 Farben (Infusionspumpe) den Wert 0.8 voraussagt.

4.4.5.4 Integration der Bewertungsverfahren in CEDI

Die Analyse von farbigen Bildschirmdarstellungen durch die Berechnung der Buntheiten kann im Farbeditor CEDI unter der Menütaste "BI" aufgerufen werden. Es kann dann interaktiv auf einem zweiten Bildschirm der zu untersuchende Bildausschnitt festgelegt werden. Neben dem Mittelwert und der Standardabweichung der Buntheit

wird die Anzahl der Farborte und die Schätzung der Bewertungen K1, K2 und K3 angegeben.

Zusätzlich werden die Farborte in die a,b-Ebene (CIELAB) projiziert dargestellt, wobei die Größe der Markierung von der relativen Häufigkeit eines Farborts im betrachteten Bildschirmausschnitt abhängt. Durch dieses Muster kann neben der Analyse der Buntheit (Farbsättigung) auch die Farbortverteilung betrachtet und beurteilt werden. So zeichnen sich die positiv beurteilten Entwürfe u.a. dadurch aus, daß die Farborte überwiegend auf einer Linie sog. Gegenfarben liegen (Bild 4.30). Diese Erfahrung stimmt mit der Beobachtung von Murch (1984) überein, daß Gegenfarben gut zueinander passen.

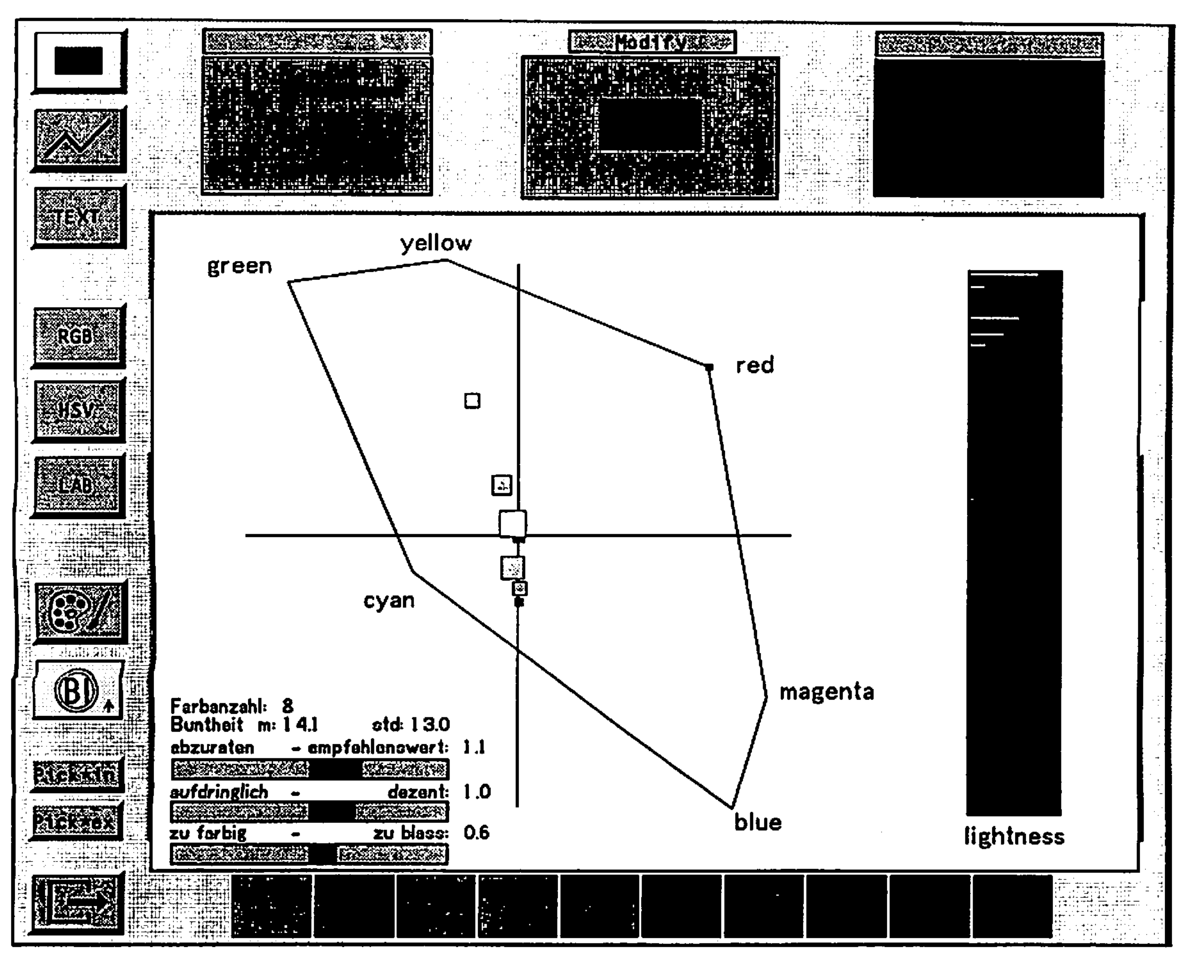


Bild 4.30: CEDI mit Analyse von Entwurf Nr. 3.

Das wesentliche Ergebnis dieses Abschnitts liegt in der Realisierung eines interaktiven Werkzeugs zur Farbgestaltung, dessen Konzept ausgehend von Perzeption und Kognition menschlicher Farbwahrnehmung entwickelt wurde. Dabei spielte die Berücksichtigung des Simultankontrastes eine hervorgehobene Rolle für die Gestaltung und Beurteilung von Bildschirmentwürfen.

Zusätzlich zur Unterstützung einer heuristischen Vorgehensweise wurde ein objektives Bewertungsverfahren entwickelt, das sich an der Buntheit der auf dem Bildschirm verwendeten Farben orientiert. Dieses Bewertungsverfahren wurde in den Funktionsumfang des Farbeditors CEDI integriert und bietet die Möglichkeit einer Voroptimierung. Obwohl der Aspekt der Buntheit nur einen Aspekt bei der Gesamtwirkung einer Farbgestaltung ausmacht, konnte an diesem Beispiel gezeigt werden, wie ausgehend von einer Hypothese über eine Studie bis hin zur Integration des Ergebnisses in ein Werkzeug vorgegangen werden kann.

5. Werkzeuge zur multimodalen Interaktion mit Einschluß des auditiven Kanals

Die Mensch-Maschine Interaktion wird durch Verwendung neuartiger taktiler, visueller und auditiver Interaktionsmöglichkeiten zwischen Mensch und Computer künftig "revolutionsartig" verändert (Foley 1987). Die Grundidee der multimodalen Interaktion liegt in der Verbreiterung des expliziten Kommunikationskanals durch Einbeziehung mehrerer Interaktionsformen. Bisher werden in konventionellen Systemen meist zwei taktile Modi (Tastatur und Mauszeiger) kombiniert; mit der Einbeziehung des auditiven Kanals besteht hingegen wenig praktische Erfahrung.

Die folgenden Abschnitte behandeln zunächst die kognitiven Grundlagen der multimodalen Interaktion und leiten dann daraus ein Konzept für Werkzeuge ab, die innerhalb der hier vorgestellten Entwicklungsumgebung den Bereich auditiver Interaktion abdecken.

5.1 Kognitive Grundlagen - Die "Multiple Resource Theory"

Norman und Bobrow (1975) entwarfen das Konzept der "Performance Resource Function" (PRF). Diese Funktion beschreibt den Zusammenhang zwischen der mentalen Belastung durch eine Aufgabe und der Leistung eines Probanden. Wenn man die PRFs von zwei gleichzeitig ausgeführten Aufgaben in einem Diagramm darstellt, so erhält man eine sog. "Performance Operating Characteristic" (POC). Bild 5.1 zeigt einen POC-Verlauf, der darauf hindeutet, daß die Art und Schwierigkeit der beiden Aufgaben A und B keine kognitiven Ressourcen freiläßt und daher die Leistung in der ersten Aufgabe reziprok zur Leistung in der zweiten verläuft.

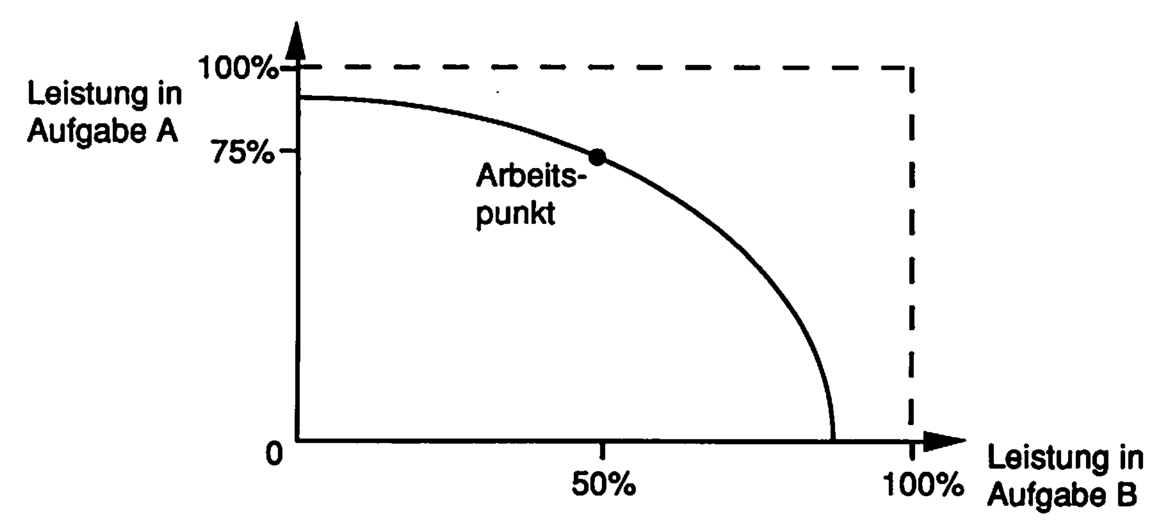


Bild 5.1: "Performance Operating Characteristic" (Wickens 1984).

Im eingezeichneten Arbeitspunkt können 75% von Aufgabe A und 50% von Aufgabe B erledigt werden. Auffallend am Diagramm ist, daß allein das Vorhandensein einer Zweitaufgabe die Erstaufgabe einschränkt. Der gestrichelte rechteckige POC-Verlauf in Bild 5.1 hingegen bedeutet, daß entweder unterschiedliche Ressourcen ausreichend für jede Aufgabe bereitstehen oder die Summe der Anforderungen an eine Ressource für die vollständige Ausführung beider Aufgaben kleiner als 100% ist.

Aus der Interpretation einer Vielzahl von POC Kurven bei unterschiedlichen Aufgabenbedingungen wurde die "Multiple Resource Theory" aufgestellt. In dieser Theorie wird die gesamte Gedächtnisleistung in einzelne "kognitive Segmente" eingeteilt (Bild 5.2). In diesen Segmenten arbeiten unabhängig voneinander sog. "Analysatoren". Dieser Ansatz geht damit über den in Abschnitt 4.1 vorgestellten Ansatz des GOMS-Modells hinaus, das nur drei seriell gekoppelte Prozessoren vorsieht. Wenn zwei Aufgaben verschiedenen Segmenten zugeordnet werden können, so ist eine besonders effiziente Bearbeitung möglich, da die Aufgaben weitgehend parallel abgearbeitet werden. Neben der Perzeption und zentralen Informationsverarbeitung ist auch die Antwortreaktion auf einen Reiz wiederum ein unabhängiger Prozeß, der in verschiedenen Segmenten ablaufen kann (Wickens und Kessel 1980).

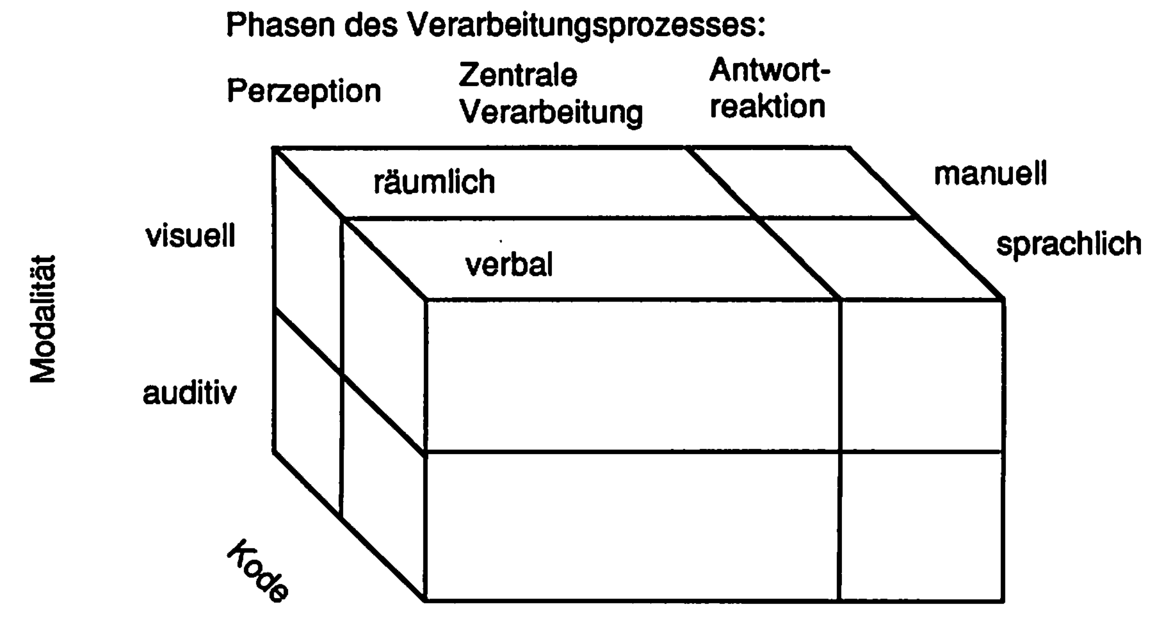


Bild 5.2: Aufteilung kognitiver Ressourcen nach Wickens (1984).

Die Informationspräsentation kann vier verschiedene Formen annehmen, die nach dem Modus ihrer Darstellung (visuell/auditiv) und der Art ihres Inhalts (verbale/räumliche Information) eingeteilt sind (Bild 5.3). Eine optimale Zuordnung ergibt sich, wenn verbale Information auditiv und räumliche Information visuell präsentiert wird.

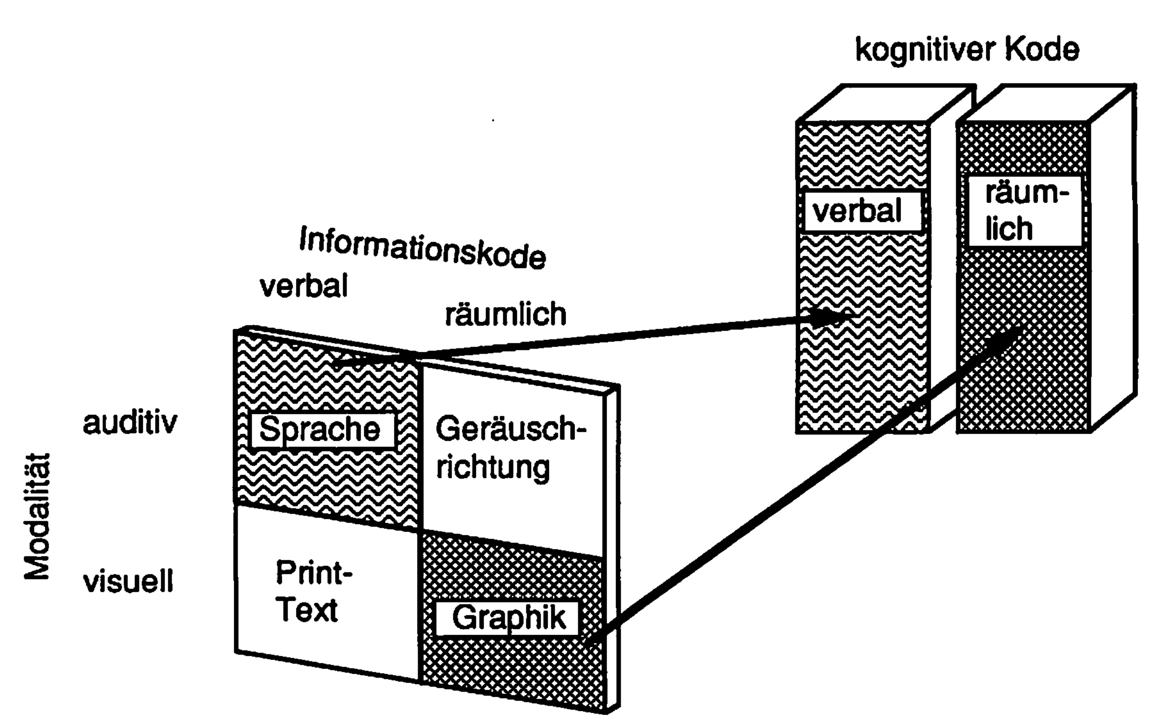


Bild 5.3: Optimale Zuordnung von Kode und Verarbeitung nach Wickens (1984).

In Bild 5.4 sind dem Verarbeitungsprozeß zusätzlich Medien der Informationspräsentation und Eingabeinteraktion zugeordnet. Die softwaretechnische Integration unterschiedlicher Interaktionsformen wurde bereits im Konzept des Interfacebaukastens berücksichtigt (siehe Abschnitt 3.2.3). Die in Bild 5.4 dargestellten Eingabe- und Ausgabekanäle werden durch dieses Konzept parallel zur Verfügung gestellt. Darüber hinaus sollen nun durch einzelne Werkzeuge der Entwicklungsumgebung die bislang wenig verwendeten Modi Spracheingabe und akustische Ausgabe effizient unterstützt werden.

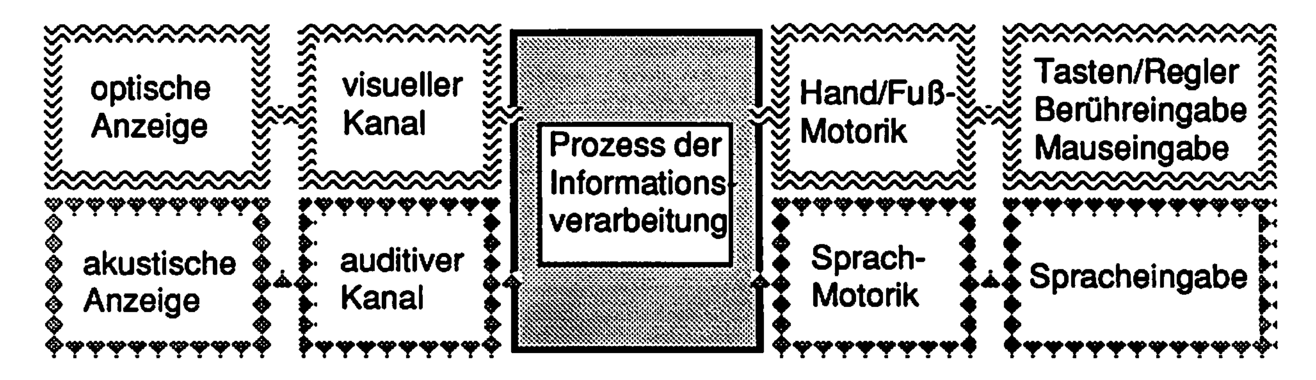


Bild 5.4: Informationsübertragung an Mensch-Maschine Schnittstellen.

5.2 Werkzeug zur Gestaltung von Spracheingabe

Der erfolgreiche Einsatz der Interaktionsform Spracheingabe in einem Informationssystem erfordert entsprechende Werkzeuge für Entwurf, Training und Bewertung eines Wortschatzes, die innerhalb der Entwicklungsumgebung zur Verfügung gestellt werden sollen.

5.2.1 Ergonomische und technische Aspekte von Spracheingabesystemen

Die Vor- und Nachteile der Spracheingabe aus ergonomischer Sicht lassen sich wie folgt zusammenfassen:
(+) Hände bleiben für parallele manuelle Aufgaben frei,
(+) Spracheingabe benötigt keinen zusätzlichen Platz am Bildschirm,
(+) Spracheingabe ist eine leicht erlernbare Interaktionsform,
(-) Erkennungsrate < 100 Prozent,
(-) Fehlerkorrektur bei Falscherkennung ist schwierig,
(-) Notwendigkeit des Wortschatztrainings für jeden Benutzer.

Man unterscheidet zwischen Geräten für Einzelworterkennung und Erkennung von kontinuierlicher Sprache. Einzelworterkennung basiert auf der Mustererkennung eines im Referenzspeicher abspeicherten Sprachmusters von bis zu 3 Sekunden Dauer. Die kommerziell verbreiteteten Systeme arbeiten überwiegend sprecherabhängig; der Umfang der gleichzeitig im Referenzspeicher liegenden Wortmuster liegt zwischen 100 und 1000. Sprecherunabhängige Worterkennung mit hoher Erkennungssicherheit ist bisher nur für die Ziffern 0..9 kommerziell verfügbar. Jedoch bieten einige sprecherabhängige Systeme die Möglichkeit, die Einzelworterkennung quasi sprecherunabhängig zu betreiben, indem ein Referenzmuster über möglichst viele Sprecher gemittelt wird. Die Erkennungssicherheit liegt dann aber niedriger als bei einem individuellen Wortschatztraining.

Systeme für kontinuierliche Spracherkennung führen eine Phonemerkennung durch, die anschließend mit syntaktischem und semantischen Wissen einer bestimmten Landessprache verknüpft wird. Derartige Systeme sind so komplex, daß sie bislang nur in Forschungslabors erprobt werden. Für die ausführliche Beschreibung von technischen Grundlagen und Verfahren wird auf Sickert (1984) verwiesen.

5.2.2 Wortschatzentwurf, Training und Erkennungsbetrieb

Für eine sprecherabhängige Einzelworterkennung müssen die im folgenden vorgestellten Phasen Wortschatzentwurf, Wortschatztraining, Wortschatztest und Erkennungsbetrieb genauer betrachtet werden.

5.2.2.1 Wortschatzentwurf

Die Wörter eines Erkennungswortschatzes sollen so gewählt werden, daß sie mit dem Vokabular des Benutzers (bzw. der Benutzergruppe) kompatibel sind, d.h. auch Fachausdrücke oder sog. "Laborjargon" müssen berücksichtigt werden. Diese Forderung kann durch eine Benutzerpartizipation in der Entwurfsphase erreicht werden. Die Worte eines Wortschatzes sollen das Kriterium der phonetischen Unterscheidbarkeit aufweisen, d.h. Worte wie "Herz" und "Schmerz" dürfen nicht in demselben Wortschatz auftreten. Bailey (1982) nennt dies einen "phonetisch ausbalancierten Wortschatz", ohne jedoch ein Verfahren oder eine quantitative Maßzahl anzugeben. Einen nicht balancierten Wortschatz erkennt man an der häufigen Verwechslung von zwei oder mehr Wörtern im Erkennungsbetrieb. Um nicht erst während des Erkennungsbetriebs diese Schwächen des Wortschatzentwurfs zu bemerken, können verschiedene Testmöglichkeiten vorgesehen werden (siehe 5.2.2.3).

Bei umfangreichen Wortschätzen (>100) kann das Training sehr lange dauern und ermüdend wirken. Gleichzeitig vergrößert sich die Erkennungszeit und Zahl der Verwechselungen von Worten. Zudem wird mit wachsender Anzahl von Worten die kognitive Verfügbarkeit der möglichen Befehle für den Benutzer eingeschränkt, d.h. er kann nicht alle möglichen Befehle im Gedächnis behalten. Ein Spracherkennungssystem kann hier die Unterstützung anbieten, einen Wortschatz entsprechend der Benutzeraufgabe hierarchisch zu strukturieren, d.h. der jeweils aktive Wortschatz ist auf die zu dieser Aufgabensituation passenden Worte beschränkt. Die Erkennungszeit und die Rate der Richtigerkennung werden dadurch positiv beeinflußt.

5.2.2.2 Wortschatztraining

Sprecherabhängige Erkennungssysteme erfordern von jedem Benutzer ein Wortschatztraining, bei dem das individuelle Sprachmuster von jedem Wort des Wortschatzes aufgenommen wird. Die Sprechvariabilität eines Benutzer schränkt die Erkennungssicherheit ein; daher wird eine gewisse Sprecherdisziplin vom Benutzer gefordert, d.h. bei Wortschatztraining und Erkennungsbetrieb soll in gleicher Art und Weise gesprochen werden. In einem im Helmholtz-Institut entwickelten Werkzeug zur Gestaltung von Spracherkennung wird dies durch die Darstellung von geeigneten Bildschirm-

masken unterstützt (Hoffmann 1987). Da die Silbenbetonung die Erkennung stark beeinflußt, kann in den Bildschirmformularen für Wortschatzentwurf und -training die gewünschte Betonung durch die Sonderzeichen "!" (stimmhaft) und "-" (stimmlos) gekennzeichnet werden (Bild 5.5).

Nr.	Wort	Betonung	Nr.	Wort	Betonung
1	Alpha	!-	2	Bravo	!!
3	Charlie	!-	4	Delta	!-
5	Echo	!-	6	Foxtrott	!!
7	Golf	!	8	Hotel	!!
9	India	!!-	10	Juliet	!-!
11	Kilo	!-	12	Lima	!-
13	Mike	!	14	November	-!-
15	Oscar	!-	16	Papa	!!

Bild 5.5: Bildschirmseite im Wortschatzentwurf.

Darüber hinaus gibt es vom Sprecher nicht zu beeinflußende Störungen wie Stimmveränderungen durch Erkältung oder Umgebungslärm, die die Erkennungsrate absenken. Von der technischen Seite her kann dieses Problem durch Mittelung der Referenzmuster über mehrere Sprachproben (ca. 2-10) verringert, bzw. es können für ein Wort mehrere (ungemittelte) Referenzmuster abgespeichert werden. So könnte z.B. je ein Referenzmuster mit frischer Stimme (Morgen) und eines mit ermüdeter Stimme (Nachmittag) abgespeichert werden und bei der Erkennung parallel verfügbar sein. Wenn das Wortschatztraining unter Einsatzbedingungen (nicht Laborbedingungen) stattfindet, können die Einflüsse von Umgebungsgeräuschen und Streß besser berücksichtigt werden.

5.2.2.3 Wortschatztest

Während des Wortschatztest sind noch keine Funktionen an die Befehle angebunden, d.h. er dient primär der Bestimmung der Erkennungsrate. Im Testbetrieb kann der Benutzer in beliebiger Reihenfolge Wörter sprechen und dann die Reaktion des Systems als "richtig", "zurückgewiesen" (das System zeigte keine Reaktion) oder "verwechselt mit Nr.:x" dokumentieren. Am Ende des Testbetriebs erhält der Benutzer eine Statistik in Form einer Tabelle, aus der die Schwächen des Wortschatzes ablesbar sind (Bild 5.6).

Einige Spracherkennungssysteme liefern zusätzlich den Kennwert der *Übereinstimmung* mit dem erkannten Wort und eine Liste der ebenfalls sehr nahe an diesem

Kennwert heranreichenden Wörter. Im Anschluß an den Testbetrieb muß bei einer Nichterkennung (Falschrückweisung) eines Wortes dieses Wort erneut trainiert werden; bei einer Falscherkennung, d.h. einer Verwechslung mit einem anderen Wort muß wenigstens eines der beiden Wörter modifiziert und neu trainiert werden. Die Modifikation kann in der Änderung der Betonung liegen (bei mehrsilbigen Worten), eine Worterweiterung (bei kurzen Worten) oder die Wahl eines Synonyms sein.

Wort: **Alpha**

Richtig erkannt: 30 mal
Falsch erkannt: 3 mal
Zurückgewiesen: 5 mal

Nr.	Wort	Falsch	Nr.	Wort	Falsch
1	-	-	2	Bravo	0
3	Charlie	0	4	Delta	1
5	Echo	0	6	Foxtrott	0
7	Golf	0	8	Hotel	0
9	India	0	10	Juliet	0
11	Kilo	0	12	Lima	0
13	Mike	0	14	November	0
15	Oscar	0	16	Papa	2

Bild 5.6: Auswertung im Testbetrieb.

5.2.2.4 Erkennungsbetrieb

Die erste Voraussetzung für einen sicheren Erkennungsbetrieb ist die Einhaltung der Bedingungen des Wortschatztrainings (Sprecherdisziplin). Dies umfaßt insbesondere die einheitliche Mikrofonplazierung, Lautstärke und Betonung. Eine andere Möglichkeit zur Verbesserung des Erkennungsbetriebes wird durch die Ein/Aus Kontrolle erreicht. Damit ein System nicht ständig im Erkennungsbetrieb ist, kann ein Umschaltmechanismus vorgesehen werden, z.B. in Form eines Fußschalters. Hier wurde eine Softwarelösung durch spezielle Sprachsignale (Befehle "Aufwachen" und "Schlafen") realisiert, um zwischen aktiven und passiven Erkennungsbetrieb umzuschalten.

Die Erkennungsrate heutiger Systeme liegt stets unter 100%, so daß es zwangläufig zu Fehlern kommt. Durch Variation des Schwellwertes für eine Richtigerkennung kann der Erkennungsbetrieb an spezielle Erfordernisse (z.B. hohe Schwelle für sicherheitskritische Befehle) angepaßt werden. Bei Nichterkennung (Falschrückweisung) muß ein Wort wiederholt werden; um Wortverwechslungen (Falschakzeptanz) abzufangen, kann ein Feedback entweder auditiv (Sprachausgabe) oder visuell erfolgen, das dann wiederum vom Benutzer bestätigt werden muß. Dieser Aufwand ist jedoch zeitraubend

und daher nur bei folgenschweren Befehlen (z.B. Löschen von Daten) sinnvoll. Eine bessere Lösung wird durch das Feedback des Systems (Benutzerführung) und die Möglichkeit einer UNDO-Funktion erreicht. Dies wird im vorliegenden Ansatz durch die Integration in den Interfacebaukasten automatisch erreicht, weil Sprachbefehle vom Interaktionsmanagement genau wie andere Eingaben gehandhabt werden (vgl. Abschnitt 3.2).

Es wurden im Laufe der Arbeiten zwei unterschiedliche Spracherkennungssysteme eingesetzt, die durch entsprechende Interaktions-Server in das Gesamtkonzept integriert wurden:

1.) Speech System III™ von Speech Design.
   Speech System hat ein aktives Vokabular von 128 Worten, die ohne die Möglichkeit zur Mittelung abgelegt werden. Im Erkennungsbetrieb werden Wortnummern ermittelt, die der Reihenfolge der Sprachmuster beim Training entsprechen. Es handelt sich um ein rechnerunabhängiges System, das durch ein festgelegtes Protokoll per Datenleitung angesprochen werden kann.
2.) Voice Navigator™ von Articulate.
   Voice Navigator hat ein aktives Vokabular von 200 Worten. Durch hierarchische Strukturierung des Wortschatzes kann der Gesamtwortschatz jedoch größer werden. Beim Worttraining kann eine Eingabe bis zu sechsmal gemittelt werden. Das System wurde für Apple Macintosh Computer entwickelt und verfügt über entsprechede menügesteuerte Software. Es besteht die Möglichkeit Rückweisungsschwellen zu modifizieren.

5.3 Werkzeuge zur Gestaltung von akustischer Ausgabe

5.3.1 Kognitive und technische Aspekte

Akustische Information ist im Gegensatz zu visueller Information transient (flüchtig). Die kognitiven Fähigkeiten des Menschen wirken hier ausgleichend, indem akustische Information 5 Sekunden, visuelle nur 0.5 Sekunden im Kurzzeitgedächnis verfügbar bleibt (Darwin et al. 1972). Eine parallele, vollständige Verarbeitung mehrerer Sprachquellen ist nicht möglich, jedoch kann eine Sprachquelle aus vielen anderen herausgefiltert werden ("Cocktailparty Effekt"). Gleichzeitig können aber auch andere Quellen "akustisch beobachtet" werden, so daß bei bestimmten Schlüsselwörtern die Aufmerksamkeit sofort auf andere Quellen umgeschaltet werden kann (Treisman 1960). Sprachinformation wird hauptsächlich im Kontext aufgenommen, so daß ein fehlendes Wort im Satz oft aus dem Kontext heraus erraten werden kann. Es gibt zwei Arten von akustischen Ausgabesystemen:

1. Reproduktive Ausgabesysteme, d.h. Wiedergabe digitalisierter Sprache.
2. Syntetische Umsetzungssysteme, z.B. Wort/Phonem-Generierung aus Text oder Klang-Generierung aus Frequenz- und Phasenspektrum.

In reproduktiven Systemen wird ein Sprachmuster digital kodiert und bei Bedarf wieder in ein analoges Signal verwandelt. Die aus Text synthetisch generierte Sprache klingt heute noch aufgrund fehlender Satzmelodie und Koartikulation (Phoneme variieren abhängig vom Kontext) unnatürlich. Synthetische Klangerzeugung wird bereits seit einigen Jahren -besonders im Bereich Musik- erfolgreich eingesetzt. In der Entwicklungsumgebung wurden für die akustische Ausgabe sowohl digitalisierte Sprache als auch synthetisch erzeugter Klang berücksichtigt.

5.3.2 Generierung und Speicherung von Sprache/Klang

Die Werkzeugumgebung für akustische Ausgaben wurde auf einem seperaten Rechner implementiert, der mit geeigneter Hard- und Software[1] für akustische Ausgabe ausgerüstet wurde. Diese Erweiterung bietet die Möglichkeit, Sprache zu digitalisieren oder Töne zu generieren und dann diese Muster interaktiv sowohl im Zeitbereich zu editieren (ausschneiden, löschen, verschieben; Dämpfung einer Hüllkurve, Echo, etc.) als auch im Frequenz und Phasenspektrum zu manipulieren (Filterung, Phasenverschiebung). Ein so erzeugtes Sprach/Klangmuster kann als "snd-Resource" (Apple 1988) gespeichert werden. Es wurde ein interaktiver Zugriff auf die gespeicherten Muster realisiert, indem diese Bibliothek mit einer Benutzeroberfläche in Form eines Notizbuches ausgestattet wurde (Bild 5.7).



Bild 5.7: Bibliothek in Form eines Notizbuches.

[1] Apple Macintosh IIx mit "MacRecorder™", Software:"SoundEdit™"

Dieses Notizbuch gibt einen Überblick über alle "snd-Resources", ohne daß das Editierprogramm benutzt werden muß. Hierzu wird zunächst eine Seite aufgeschlagen (z.B. I-L), indem die entsprechede Registerlasche angeklickt wird. Die aufgeschlagene Seite (Bild 5.8) besitzt interaktive Tasten zum Abspielen der entsprechenden akustischen Meldungen; durch Anklicken einer Taste, z.B. Lasix, wird die dazugehörige "snd-Resource" abgespielt.

Bild 5.8: Aufgeschlagene Seite des "Notizbuches".

Das Hinzufügen neuer Tasten mit neuen "snd-Resources" ist sehr einfach; durch Drücken der Taste "wie trage ich einen neuen Sound ein" (siehe Bild 5.7) wird dieser Vorgang auf Wunsch erklärt. Hilfsprogramme, wie z.B. der Aufruf des Editors, sind ebenfalls in dieser Umgebung integriert.

Es wurde bereits erwähnt, daß eine "snd-Resource" sowohl ein Klang oder eine Klangfolge als auch ein Wort oder eine Wortkette sein kann. Daraus ergeben sich zwei verschiedene Vorgehensweisen zum Aufbau einer Bibliothek:

- Jede applikationsspezifische akustische Nachricht wird unter einem Namen abgelegt.
- Jedes einzelne Wort (Klang) wird unter einem Namen abgelegt, und erst später individuell zu einer Nachricht zusammengesetzt.

Beide Varianten wurden für die Applikation "Infusionspumpen-Kontrolleinheit" eingesetzt und getestet. Die Alarmmeldungen für eine unterbrochene oder abgeklemmte Zuleitung sollten in der Form:

"Alarmton1 Die Zuleitung von MEDIKAMENT ist offen",

"Alarmton2 Die Zuleitung von MEDIKAMENT ist abgeklemmt"

erscheinen, wobei für MEDIKAMENT ca. 25 verschiedene Namen möglich sind. Die

erste Vorgehensweise führt zu einer sehr großen Anzahl von langen Meldungen (bei 25 Medikamenten 50 verschiedene Meldungen). Die zweite Vorgehensweise benötigt weit weniger Sprachmeldungen, führte jedoch zu einer unnatürlichen Satzmelodie. Es bewährte sich schließlich ein Kompromiß beider Vorgehensweisen: Die Worte "Die Zuleitung von", "ist offen" und "ist abgeklemmt" wurden jeweils als Einheit abgelegt. Hinzu kamen die 25 Medikamentennamen sowie die beiden Alarmtöne, so daß insgesamt nur 30 relativ kurze Meldungen für die genannte Aufgabe in der Bibliothek abgelegt wurden.

Die Integration der Sprachausgabe in den Interfacebaukasten wurde bereits in Abschnitt 3.2.2 in Zusammenhang mit dem Objekt "Speaker" angesprochen. Für die akustischen Ausgabe können zwei verschiedene Objekt-Methoden verwendet werden:
1. play: Ausgabe einzelner "snd-Resources" durch Angabe ihres Namens.
2. playText: Ausgabe einer Folge von Ressourcen, wobei jedes Wort einer Wortkette nacheinander als Argument von "play" eingesetzt wird.

Die Argumente der Methode "play" werden als ASCII-String auf eine serielle Schnittstelle geschrieben, und vom Empfängerprogramm (=Sprachausgabeprogramm) als Name einer "snd-Resource" interpretiert und abgespielt.

5.3.3 Bewertung von akustische Meldungen

Bei der Gestaltung synthetischer Signale stehen dem Entwickler viele Freiheitsgrade offen. Daher ist - ähnlich wie bei Piktogrammen - eine Bewertung durch den Benutzer erforderlich, um optimale Ergebnisse zu erreichen. Besonders kritisch ist die Gestaltung von akustischen Alarmsignalen; eine Studie von Stanford (1988) zeigte, daß die Alarmsignale kommerzieller Monitoring-Geräte von Anästhesisten als negativ beurteilt wurden. Dies ist eine Erklärung dafür, daß die akustischen Alarme an diesen Geräten oft völlig abgestellt werden.

Als Konsequenz daraus wurde in der Entwicklungsumgebung die Möglichkeit geschaffen, "snd-Resources" in Form eines Fragebogens interaktiv am Bildschirm zu bewerten. Dieser "elektronische Fragebogen" wurde mit Hilfe von "HyperCard" (Apple 1988) realisiert. Bild 5.9 zeigt eine für die Bewertung einer akustischen Meldung gestaltete Karteikarte. Während diese Karte ausgefüllt wird, kann die zu bewertende Meldung durch Anklicken der Sprechblase beliebig oft abgespielt werden. Die Versuchsperson klassifiziert die Meldung zunächst als "Alarm" oder "Information". Die in der Beurteilung verwendeten Adjektive sind der "Mood Adjective Check List" entlehnt, einem Standardinstrument aus der experimentellen Psychologie, das auch von Stanford (1988) eingesetzt wurde.

3. Beurteilung der Meldungen   Hilfe

Meldung Nr: 1
von: 3

play it, Sam!

Klassifikation: ○ Alarm ○ Information

Beurteilung:

energisch	○	◉	○	○	○	nicht energisch
unfreundlich	○	○	◉	○	○	freundlich
störend	○	○	○	◉	○	nicht störend
aggressiv	○	○	◉	○	○	nicht aggressiv
ermüdend	○	○	○	○	◉	aufmerksam machend
beängstigend	○	○	◉	○	○	beruhigend

Assoziation: ____________

Bild 5.9: Bewertung eines Sprach/Klangmusters (nach Standford 1988).

Durch Anklicken eines Kreises wählt die Versuchsperson einen Wert auf der 5-stufigen Bewertungsskala aus. Es besteht die Möglichkeit, Freitext in der Zeile "Assoziation:" mit Hilfe der Tastatur einzutragen. Jede Meldung besitzt eine eigene Karteikarte, die mit der jeweils vorigen und nächsten Karte durch die Pfeiltasten verbunden ist. Die Auswertung dieser Fragebögen erfolgt dann wieder entsprechend der in Abschnitt 4.3.3 beschriebenen Vorgehensweise für die Bewertung von Piktogrammen.

5.4 Gestaltungsprinzipien zum Einsatz multimodaler Interaktion

Bei Überwachungs- und Steuerungsaufgaben müssen die Informationen der Ein- und Ausgabe geeignet auf den visuellen, auditiven und taktilen Kanal aufgeteilt werden. Eine kontinuierliche Überwachung mehrerer parallel dargestellter Informationsquellen ist nur visuell möglich. Daher soll für Überwachungsaufgaben nur in Ausnahmefällen (z.B bei Überlastung des visuellen Kanals) auf den auditiven Kanal ausgewichen werden. Die akustische Ausgabe kann auch als zusätzliche redundante Informationsquelle für die auf dem Bildschirm dargestellte Information dienen.

Warnungen werden besser in Form eines akustischen Signals wahrgenommen, weil selten auftretende Signale eher in einem Kanal wahrgenommen werden, der *im* Gegensatz zum visuellen Kanal nicht permanent belegt wird (Posner et. al. 1976); ein Laut kann nicht so leicht "übersehen" werden wie ein visueller Reiz und provoziert im

Vergleich zu visuell dargebotenen Warnungen schnellere Reaktionen (Simpson 1987). Die akustische Information soll nicht als isoliertes Schlüsselwort, sondern innerhalb eines redundanten Satzkontextes angeboten werden. Die Wirkung eines Alarms kann durch einen einleitenden charakteristischen Klang oder durch Wiederholung der Meldung erhöht werden.

Den genannten Prinzipien entsprechend wurden für die Infusionspumpen-Kontrolleinheit folgende 3 Alarmmeldungen realisiert:

"Alarmton1 Die Zuleitung von MEDIKAMENT ist offen",
"Alarmton2 Die Zuleitung von MEDIKAMENT ist abgeklemmt"
"Alarmton3 Die Infusionspumpe mit MEDIKAMENT ist leer"

Die Qualität eines Interaktionsmodus kann nur abhängig von den Eigenheiten einer damit bearbeiteten Aufgabe (verbal und räumlich) beurteilt werden (Reiz-Verarbeitungs-Reaktions Kompatibilität). So könnte z.B. die Anweisung zur Auswahl eines Ventils akustisch, die Aktion "Öffnen" dann manuell erfolgen. Steuerungsaufgaben, die seriell in der Art einer Kommandosequenz ablaufen, können durch Spracheingabe ausgeführt werden. Auch die Abfrage bestimmter Informationen (Report) gehört in diese Kategorie. Analoge Dateneingaben, wie das Regeln eines Potentiometers, können nur sehr schlecht durch Spracheingabe vorgenommen werden.

Für einen qualitativen Vergleich werden folgende Interaktionen betrachtet (Foley 1984):

- Selektion
- Positionieren
- Quantifizieren
- Texteingabe

	Selektion	Positionieren	Quantifizieren	Texteingabe
Berührung	++	+	+	-
Mauseingabe	+	++	+	-
Tastatur	-	-	+-	+
Sprache	++	-	-	++

Tab. 5.1: Eignung verschiedener Eingabemodi für bestimmte Aufgabenstellungen.

Tabelle 5.1 gibt eine Übersicht, wie gut die aufgeführten Interaktionsmodi für die einzelnen Aufgaben geeignet sind.

Als Anwendungsbeispiel für multimodale Interaktion wird der in Abschnitt 4.2 vorgestellte interaktive Entwurfseditor herangezogen; dieser Editor kann Eingaben durch Maus-, Tastatur- und Spracheingabe verarbeiten. Das Positionieren von Objekten am Bildschirm erfolgt mit Hilfe der Maus, weil nur dieser Eingabekanal eine pixelgenaue Auflösung bietet. Die Auswahl eines Menüpunktes (Selektion) kann sowohl per Maus als auch per Spracheingabe vorgenommen werden, so daß hier zwei Eingabekanäle parallel zur Verfügung stehen. Die Kombination von Selektion durch Sprache und Positionierung durch Maus ist besonders gut für den Entwurf auf einem seperaten Bildschirm geeignet, weil in diesem Fall die Darstellung der Menüauswahl vom Entwurfsbildschirm getrennt ist. Die Tastatur wird z.T. für das Ausfüllen der Attributformulare benötigt; Spracheingabe wird hierfür nicht vorgesehen, weil die Vielzahl möglicher Attribute den erforderlichen Wortschatz erheblich vergrößern würde.

Mit der Beschreibung der Werkzeuge für Spracheingabe und akustische Ausgabe ist nun das Konzept der in Kapitel 2 vorgestellten Entwicklungsumgebung vollständig erläutert worden. Die Funktionalität der Werkzeuge wurde dabei meist am Beispiel der Infusionspumpen-Kontrolleinheit vorgestellt. Im folgenden Kapitel wird abschließend die Gestaltung einer komplexeren Benutzerschnittstelle aus dem Bereich der Patientenüberwachung in der Anästhesiologie beschrieben.

6 Gestaltung der Benutzerschnittstelle exemplarisch an einem Entscheidungs-unterstützenden System in der Medizin

Bei der Entwicklung eines Entscheidungs-unterstützenden Systems werden hohe Anforderungen an die Gestaltung der Benutzerschnittstelle gestellt, weil dem Benutzer bei diesen Systemen sehr komplexe Zusammenhänge in verdichteter Form veranschaulicht werden müssen. Im medizinischen Bereich kann die Motivation für ein Expertensystem wie folgt zusammengefaßt werden (Haux 1987):

- die Genauigkeit der klinischen Diagnose zu erhöhen,
- die Zuverlässigkeit der Diagnose zu verbessern,
- die Effizienz von Untersuchungen und Therapien zu verbessern und
- das Verständnis der Struktur medizinischen Wissens zu erhöhen.

Für den Arzt als Benutzer eines solchen Systems stehen folgende Fragen im Vordergrund:

- Ist das System wirklich hilfreich (sind Routineaufgaben schneller oder schwierige Entscheidungen mit wenig Zeitaufwand besser lösbar) ?
- Ist das System zuverlässig ?
- Ist es leicht zu bedienen, die Antwortzeit kurz ?
- Hilft es, ohne bestimmend zu sein ?
- Kann es Empfehlungen rechtfertigen, so daß man auch für sich alleine entscheiden kann ?

Die Benutzerschnittstelle soll entsprechend einem systemergonomischen Ansatz so gestaltet werden, daß das Gesamtsystem, bestehend aus Entscheidungs-unterstützendem System und Arzt als Benutzer, in bezug auf die Aufgabe der Patientenüberwachung optimiert wird.

In diesem Kapitel wird beschrieben, wie ausgehend von der Aufgabenanalyse die Gestaltung von Informationsdarstellung und Interaktion in einem Entscheidungs-unterstützenden System entwickelt werden kann. Bei der Gestaltung kommen die zuvor beschriebenen Entwicklungswerkzeuge zum Einsatz, die insbesondere durch Visualisierungstechniken den Informationsfluß zwischen Arzt und Computersystem unterstützen. Hierbei unterstützt das Computersystem die Beurteilung des Patientenzustands und gibt Vorschläge zur Therapie, während Aktionen am Patienten (Medikamentengaben) Aufgabe des Arztes bleiben.
Zunächst wird das menschliche Entscheidungsverhalten behandelt, das gerade für Entscheidungshilfen in komplexen Situationen soweit wie möglich bekannt sein und berücksichtigt werden soll.

6.1 Kognitives Verhalten bei der Entscheidungsfindung

Die kognitiven Fähigkeiten des Menschen beschränken ihn bei der Entscheidungsfindung auf das Abwägen innerhalb einer sehr kleinen Anzahl von Hypothesen, d.h. auch hier ist Miller's Gesetz "7 ± 2" (Miller 1956) gültig. Wenn mehr Information angeboten wird als verarbeitet werden kann, so erschwert diese kognitive Überlastung die Entscheidungsfindung und führt zu einer weniger optimalen Entscheidung. Im Entscheidungsprozeß werden einfache "Daumenregeln" bevorzugt, um die Menge auf 4-6 Hypothesen einzugrenzen, selbst wenn effiziente, aber kognitiv anspruchsvollere Alternativen zur Verfügung stehen. Bei starker Beanspruchung (Streß) wird die Flexibilität menschlichen Denkens und Handelns am stärksten eingeschränkt, es tritt der Effekt "cognitive tunnel vision" auf, d.h. ein Abwägen der verfügbaren Informationen oder eine Änderung des einmal begonnenen Gedankengangs wird unmöglich (Sheridan 1981, Moray 1981).

6.1.1 Unterstützung kognitiver Prozesse

Ein Hauptanliegen der Entscheidungshilfen ist die Strukturierung der Vielfalt von Informationen und Entscheidungsalternativen. Ein intelligentes Entscheidungsunterstützungssystem soll folgende wichtige Anforderungen erfüllen (Norman 1986):

- es soll dem Benutzer helfen, den aktuellen Status fortlaufend zu erfassen und bei der Diagnose Flexibilität zu bewahren (Vermeidung von "cognitive tunnel vision"),
- der Benutzer soll in die Prozesse von Beobachtung und Steuerung ("control loop") eingeschlossen sein. Anstatt nur passiver Beobachter zu sein, soll er jederzeit eine aktive Rolle einnehmen können,
- in der verfügbaren Informationsmenge sollen Prioritäten gesetzt werden. Das Hauptaugenmerk ("major focus") soll stets auf das größte Problem gelenkt werden.

Zur Verwirklichung dieser Anforderungen formuliert Kraiss (1986) u.a. folgende Gestaltungsprinzipien für kognitive Unterstützungsfunktionen:

1. Reduzierung der Informationsmenge durch Informationsvorverarbeitung, -filterung und -strukturierung.
2. Gedächtnisentlastung durch Visualisierung von Information. Die Bandbreite des Informationsflusses an der Mensch-Maschine Schnittstelle soll durch den Einsatz neuartiger Interaktions- und Gestaltungsprinzipien verbessert werden.
3. Vereinfachung des Entscheidungsprozesses durch Darbietung vorab bewerteter Alternativen.

zu 1.: Die Strukturierung von Information kann durch Abstraktion und Aggregation von Daten erreicht werden (Rasmussen 1986). Es entstehen Funktions- und Informationsebenen, die die Berücksichtigung der unterschiedlichen Relevanz von Daten für einen Problemlösungsprozeß durch verschiedene Hierarchiestufen ermöglichen (Woods 1986). Durch Aggregation werden Informationen gebündelt, so daß diese Zusammenfassung in höherwertige Informationseinheiten (Chunks) dem Entscheidungssuchenden erlaubt, trotz der o.g. Einschränkungen des Kurzzeitgedächtnis eine Vielzahl von Informationen zu berücksichtigen ("decision complexity advantage"). Informations-Chunks sind eine Möglichkeit, die Vielzahl von Meßanzeigen zu reduzieren, die bei der "ein Sensor => ein Indikator" Technik entsteht. Die Fähigkeit, eine große Anzahl von Informationen durch eine geeignete Struktur auf eine bearbeitbare Größe zu komprimieren, wird als typisches "Expertenverhalten" bezeichnet (Anderson 1981).

zu 2.: Neben der Strukturierung von Information ist die Visualisierung von Daten und deren Beziehung untereinander eine effektive Möglichkeit menschliches Problemlöseverhalten zu verbessern. Das visuelle System ist evolutionsbedingt besonders leistungsfähig. Eine geeignete räumliche Struktur einer Anzeigetafel, die einen Prozess adäquat widerspiegelt, kann die hohe mentale Leistung der sensormotorischen Verarbeitung direkt für die Überwachungsaufgabe nutzen. Die kognitive Distanz zu komplexen Prozessen, die heute indirekt über Sensoren und Stellglieder erfaßt werden, wird durch eine piktografische Repräsentation verkürzt.

zu 3.: Die Darbietung von verschiedenen Alternativen bei einer Entscheidung bewirkt, daß der Benutzer nicht auf einen Lösungsaspekt hin eingeschränkt wird, sondern zwischen verschiedenen Möglichkeiten abwägen kann. Dadurch wird seine aktive Rolle innerhalb des Gesamtprozesses unterstützt.

6.1.2 Abfolge einer Entscheidungsfindung

Die Überwachungsaufgabe bei komplexen Systemen kann in verschiedene Phasen eingeteilt werden. Zunächst wird durch Beobachten (Monitoring) entschieden, ob sich das System fehlerfrei (normal) verhält oder nicht. Im Falle einer Abweichung vom geforderten Zustand muß aufgrund einer Inspektion eine Diagnose gestellt und anschließend eine geeignete Gegenmaßnahme geplant und ausgeführt werden.

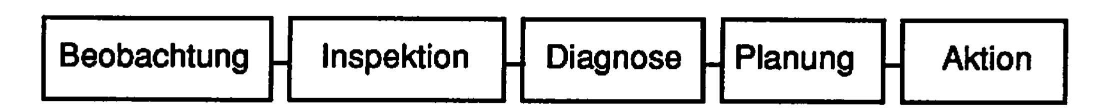


Bild 6.1: Aufgabenabfolge bei einer Überwachungsaufgabe.

Diese typische Abfolge wurde von Rasmussen (1976) genauer analysiert. Rasmussen unterscheidet drei charakteristische Verhaltensstufen, die gleichzeitig eine Gliederung der kognitiven Anstrengung zwischen Beobachtung und Aktion darstellen:

- auf Fertigkeiten basierendes Verhalten (skill-based behaviour):
  dies sind quasiautomatisch ausgeführte Verhaltenssequenzen ohne bewußte Kontrolle, z.B. geübtes Geigespielen, Radfahren etc.; Fehler treten aufgrund einer kurzfristige, topographischen Desorientierung auf (engl.: slip)
- regelbasiertes Verhalten (rule-based behaviour):
  Das Entscheidungsverhalten folgt erlernten "wenn-dann" Regeln oder bekannten Mustern, die assoziativ mit einer Situation verknüpft sind. Im Gegensatz zu den Fertigkeiten kann dieses Verhalten formuliert werden. Fehler treten typischer Weise als Unterlassung auf (engl.: lapses), z.B. die Notwendigkeit zum Öffnen eines Ventils war zwar bekannt, es wurde aber nicht reagiert.
- wissensbasiertes Verhalten (knowledge-based behaviour):
  wenn ein Problem nicht mit bekannten Methoden gelöst werden kann, muß eine Situationsanalyse unter Berücksichtigung der Zielvorgabe durchgeführt werden. Dabei werden neue Methoden aufgrund des zur Verfügung stehenden Wissens erarbeitet. Fehler treten auf, wenn größere kausale Zusammenhänge nicht oder falsch bekannt sind (engl. mistake).

In der Realität sind die Übergänge zwischen diesen Verhaltensstufen situationsabhängig und in der Regel fließend. Die Zeit zwischen der Detektion einer Abweichung und Ausführung einer Aktion ist umgekehrt proportional zur Vertrautheit oder Klarheit einer Situation. Während der "Neuling" eine vollständige Analyse der Situation durchführen muß, kann der "Experte" sofort nach einem Alarm die geeignete Gegenmaßnahme ergreifen. Es ist daher sinnvoll, dem Benutzer die Möglichkeit zu geben, während der Überwachung alle 3 kognitiven Stufen benutzen zu können und auf jeder Stufe entsprechend angepaßte Informationen zu erhalten; auch der Experte benötigt unter bestimmten Bedingungen Informationen, die primär für einen Neuling bestimmt sind.

6.2 Aufgabenanalyse und kognitives Modell

Der Entwurf der Benutzerschnittstelle des Anästhesie Entscheidungs-Unterstützungssystems AES-2 wurde als objektorientierter Systementwurf im evolutionären Prototyping durchgeführt. Die Methode des objektorientierten Designs (OOD) beruht darauf, eine Repräsentation des Problems aus der Realwelt in die Software-Domäne abzubilden. Unter Anwendung der objektorientierten Programmierung soll die Beschreibung des Gesamtsystems durch Objekte und Aktionen schrittweise in ein

lauffähiges Programm umgesetzt werden. Diese Methode kann bereits in der Anforderungsanalyse angesetzt werden, indem aus einer verbalen Beschreibung des Gesamtsystems eine Objekt-Aktion Struktur abgeleitet wird.

Ausgehend von der Gesamtaufgabe, der Überwachung des Patientenzustands, wird das System "Patient" hierarchisch zerlegt. Der Patientenzustand wird in die vier physiologischen Teilsysteme kardio-vaskuläres System, Nierensystem, Zentralnervensystem und respiratorisches System aufgeteilt. Der Zustand dieser Teilsysteme ist von verschiedenen Größen beeinflußt, die in bezug auf die Teilsysteme eine Hierarchiestufe tiefer eingeordnet werden können:

- kardio-vaskuläres System (Hämodynamik): Gefäßtonus, Herzleistung, Blutvolumen.
- Nierensystem: Flüßigkeitsbilanz, Elektrolythaushalt.
- respiratorisches System: Partialdrücke (CO2, O2), ph-Wert.
- Zentralnervensystem: Narkosetiefe, Regelung der Körpertemperatur.

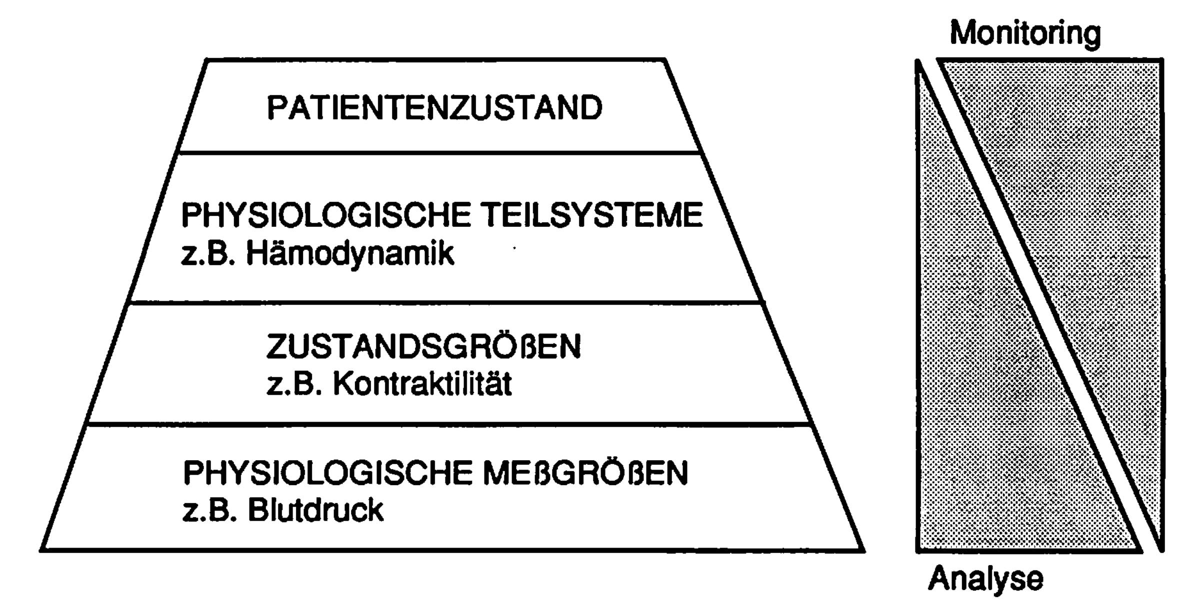


Bild 6.2: System Patient mit Teilsystem Hämodynamik.

Die Hierarchie bezüglich des kardio-vaskulären Systems ist in Bild 6.2 dargestellt. Der Aspekt der Überwachung (Monitoring) zielt auf die Frage, ob der Patientenzustand bzw. dessen Teilsysteme "normal" sind und kann besser anhand weniger zusammengefaßter Größen erfaßt werden. Im Falle einer Abweichung hingegen muß eine Analyse der Ursachen vorgenommen werden, die auf der Ebene der Vielzahl von Meßgrößen die wertvollsten Hinweise liefert.

Die genannten Teilsysteme sind für den Anästhesisten nicht in jeder Operationsphase gleichbedeutend. Die AES-2 Wissensbasis wurde für die Beurteilung des kardiovaskulären Teilsystems (Hämodynamik) in der Phase nach Abkopplung der Herz-Lungen-Maschine entworfen. Zur Beurteilung der Hämodynamik wurden 5 Zustandsgrößen eingeführt: Kontraktilität, Herzfrequenz, Vasotonus, Intravasalvolumen und Narkosetiefe. Diese Zustandsgrößen sind in der Regel nicht direkt meßbar, sondern ergeben sich aus komplexen Verknüpfungen verschiedener Meßgrößen. Dadurch kann eine Vielzahl von Einzelinformationen in einer Zustandsgröße verdichtet werden.

Die Benutzerschnittstelle des Anästhesie Entscheidungsunterstützungs Systems ist eine zusätzliche Komponente zu dem am Helmholtz-Institut enwickelten Anästhesie-Informationssystem AIS. Daher bestand die Anforderung, die Interaktion konsistent zur AIS-Schnittstelle zu entwerfen und zusätzlich neue Gestaltungsmöglichkeiten für den Bereich der Diagnosefindung und Therapieplanung einzusetzen. Der Ansatz für die Gestaltung des AES-2 besteht nun darin, aus der Aufgabenanalyse die kognitive Struktur der Aufgabe des Anästhesisten analog zum vorgestellten Modell von Rasmussen zu erfassen und die dazugehörigen Informationen geeignet strukturiert auf dem Bildschirm darzustellen.

Die Benutzerschnittstelle soll als "task-map" (Easterby 1984) strukturiert werden, d.h. Information wird entsprechend der funktionalen Zusammengehörigkeit, der Abfolge und Häufigkeit der Benutzung und der Wichtigkeit gruppiert.

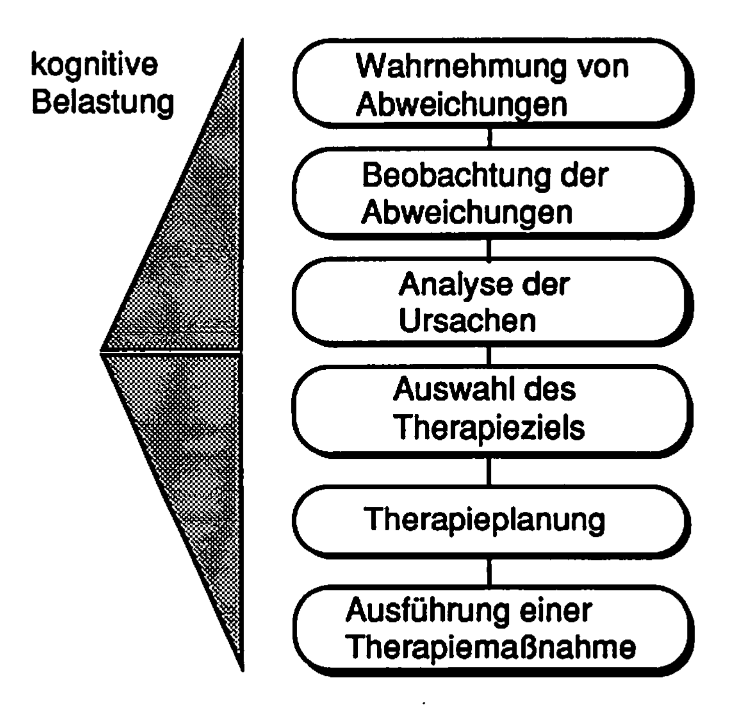


Bild 6.3: Aufgabenabfolge des Anästhesisten.

Die Aufgabe des Anästhesisten wurde in die in Bild 6.3 dargestellten Phasen eingeteilt; die kognitive Belastung steigt bis zur Auswahl des Therapieziels an und sinkt dann wieder bis die Therapie ausgeführt ist. Das Ziel der Gestaltung der Benutzerschnittstelle besteht nun darin, die kognitive Belastung in jeder Phase durch geeignete Visualisierung der relevanten Information zu verringern.

6.3 Beschreibung der Benutzerschnittstelle

In diesem Abschnitt wird ausgehend von dem im letzten Abschnitt beschriebenen kognitiven Modell des Entscheidungsprozesses die Visualisierung der notwendigen Information an der Benutzerschnittstelle beschrieben. Dabei soll die Informationsstruktur am Bildschirm an die Struktur des Entscheidungsprozesses angepaßt werden.

6.3.1 Struktur und Inhalt der Informationsdarstellung

Die Benutzerschnittstelle wird visuell durch verschiedene Informationsrahmen (Interfacebaukasten-"Frames") strukturiert. Folgende Rahmen wurden auf dem Bildschirm plaziert:

(A) Zustandsgrößen Monitoring: Bewertung der Zustandsgrößen und Trendverlauf,
(B) Meßgrößen-Daten: Werte und Bewertungen physiologischer Meßgrößen (Blutdruck, Puls, LAP),
(C) Anamnesedaten des Patienten,
(D) Therapieempfehlung: Medikament und Dosis,
(E) Therapiedokumentation: Tabelle der Therapiemaßnahmen,
(F) andere Informationen: z.B. Hinweise auf Routinemaßnahmen,
(G) Identifikation des Teilsystems, hier Hämodynamik.

Von diesen Informationsrahmen sind nur zwei in jeder Situation unverzichtbar (A und F) und müssen daher permanent sichtbar sein. Im Unterschied dazu können die anderen Informationsrahmen durch Benutzerinteraktion geöffnet und geschlossen werden, so daß der Benutzer seinen Informationsbedarf an die jeweilige Situation anpassen kann. Die räumliche Kodierung wurde so gewählt, daß jeweils benachbarte Rahmen Informationen enthalten, die auch von der Aufgabe her in enger Beziehung zueinander stehen.

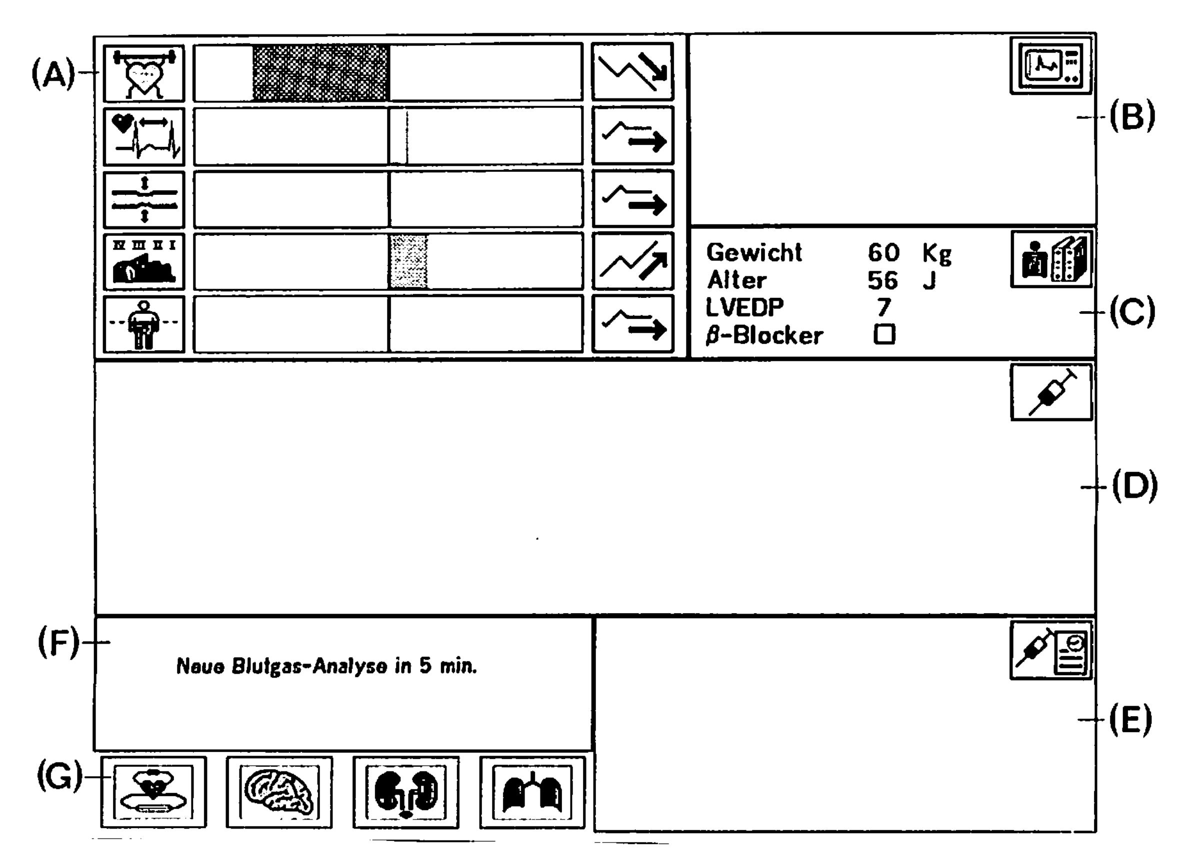


Bild 6.4: Bildschirmstruktur des AES-2.

6.3.1.1 Zustandsgrößen Monitoring (A)

Psychologische Untersuchungen haben ergeben, daß dem oberen linken Quadranten des Bildschirms die höchste Aufmerksamkeit (40%) geschenkt wird (Staufer 1987). Die Abweichung der Zustandsgrößen vom Normalzustand steht im vorliegenden Ansatz im Mittelpunkt und ist gleichzeitig Ausgangspunkt für die in Bild 6.3 dargestellte kognitive Sequenz.

Die 5 Zustandsgrößen werden durch Piktogramme visualisiert, die in der in Abschnitt 4.3.3 beschriebenen Studie entworfen und bewertet wurden. Die quasikontinuierliche Beurteilung der Zugehörigkeit einer Zustandsgröße zu den Zuständen "normal", "unter normal" oder "über normal" wird durch Anwendung der Fuzzy-Set Theorie vorgenommen. Eine geeignete Form der Darstellung der Bewertung der Zustandsgrößen sollte die kognitive Anstrengung während des Monitoring vermindern. Es wurde eine redundante Farb- und Formkodierung gewählt, da eine Kode-Kombination aus Farbe und Größe die kürzeste Suchzeit verspricht (Chase 1986).

Abweichungen vom Normalzustand werden als horizontale Balken dargestellt, die neben dem jeweiligen Piktogramm erscheinen. Ausgehend von einer Mittellinie werden nach links Zustände unter normal und nach rechts Zustände über normal dargestellt.

Proportional zum Grad der Abweichung wird die Länge des Balkens vergrößert während gleichzeitig die Farbe dieser Balken von einem grünlichen Farbton bis hin zu Rot verändert wird (siehe Abschnitt 4.4.4). Durch diese Zuordnung wird ein vertikales Profil ausgeprägt, das die menschliche Fähigkeit zur Erkennung von Bildmustern effektiver nutzen kann. Bertin (1982) stellte fest, daß "in der Senkrechten variierende Profile besser im Gedächtnis gespeichert werden als solche, die in der Waagerechten variieren". Nach einer gewissen Trainingszeit kann durch diese visuelle Unterstützung auch der komplexe Vorgang der Entscheidungsfindung auf einer eher intuitiven Ebene unterstützt werden. Mustererkennung spielt eine entscheidende Rolle in der Diagnosefindung, da in effektiver Weise vertraute Systemzustände und Störungen identifiziert werden können (Rasmussen 1986).

Das Erkennen von Trends ist ein wichtiger Faktor für die Diagnose und Therapieplanung. Im AES-2 wird daher der jeweils aktuelle Trend bestimmt und durch Piktogramme auf den Tasten rechts neben der aktuellen Bewertung dargestellt. Es erscheint eines von drei unterschiedlichen Piktogrammen abhängig davon, ob ein Wert steigt, fällt oder konstant bleibt. Durch Aktivierung einer Trendtaste wird anstatt des horizontal ausgerichteten Bewertungspegels eine Reihe senkrecht nebeneinander angeordneter Pegelanzeigen dargestellt, die die letzten 10 Bewertungen - dies entspricht einem Zeitraum von 20 Minuten - im zeitlichen Verlauf wiedergeben (Bild 6.5). Dies bietet einen detaillierteren Überblick über den zeitlichen Verlauf gegenüber dem Tastenpiktogramm, das nur eine dreistufige Quantisierung berücksichtigt.

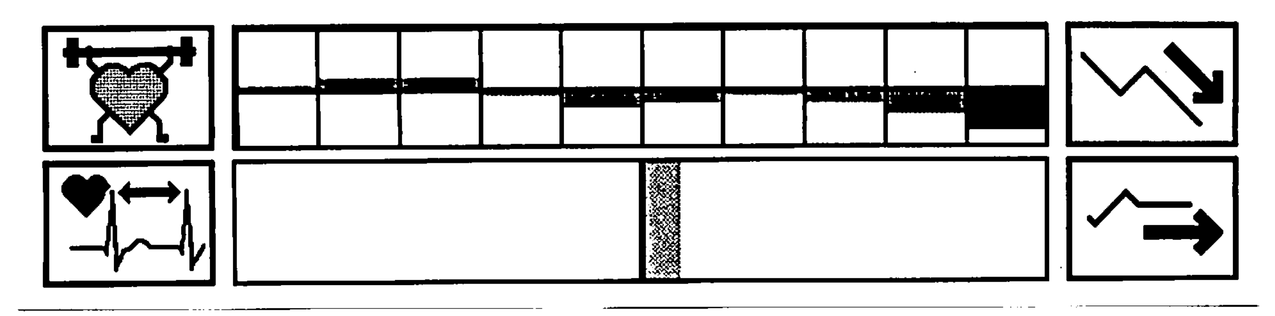


Bild 6.5: Trenddarstellung von Zustandsgrößen.

6.3.1.2 Darstellung der Meßgrößen (B)

Die Bewertung der Zustandsgrößen basiert überwiegend auf der Bewertung verschiedener physiologischer Meßgrößen, die in dem Rahmen direkt rechts neben den Zustandsgrößen dargestellt werden. Die Bewertungen der Meßgrößen Puls, systolischer Blutdruck und Blutdruck im linken Vorhof (LAP=Left Atrial Pressure) werden wie die Zustandsgrößen durch eine farbige Pegelanzeige visualisiert (Bild 6.6). Zusätzlich wird der aktuelle numerische Wert der Meßgröße dargestellt. Die benutzten

Abkürzungen und der Verzicht auf die Darstellung von Einheiten entsprechen der Vorstellung des Anästhesisten.

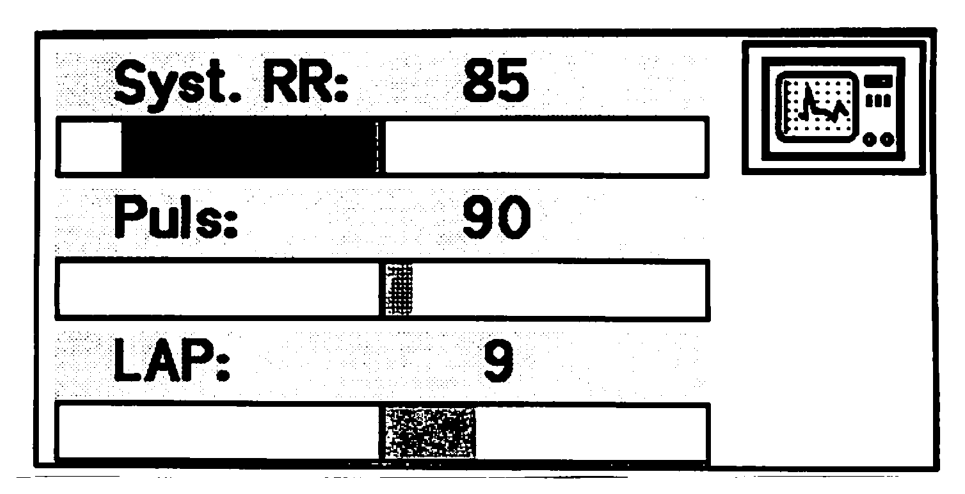


Bild 6.6: Darstellung der Meßgrößen.

6.3.1.3 Patientendaten (C)

Die Anamnese-Daten gehen sowohl in die Bewertung der Zustandsgrößen als auch in die Therapieplanung ein. Daher wird ein schneller Überblick über die wichtigsten Größen wie Gewicht und Alter des Patienten gegeben. Der präoperativ gemessene LVEDP (linksventrikulärer enddiastolischer Druck) wird zur Beurteilung des LAP herangezogen. Zur Beurteilung der Herzfrequenz ist wichtig, ob der Patient mit Beta-Blockern behandelt wurde.

Gewicht	60	Kg
Alter	56	J
LVEDP	7	
β-Blocker	☐	

Bild 6.7: Darstellung von Patientendaten.

6.3.1.4 Therapieempfehlung (D)

In der Aufgabensequenz des Anästhesisten folgt nach der Diagnose die Planung einer geeigneten Therapiemaßnahme. Die Therapieempfehlung wird in der Mitte des Bildschirms dargestellt, weil dieser Bereich besonders bei starkem Streß für den wichtigsten gehalten wird (Broadbent 1971).

Bei Aktivierung der Piktogrammtaste für den Therapievorschlag wird ein Medikament mit einer bestimmten Dosierung empfohlen (Bild 6.8). Dieses Medikament wirkt auf die Zustandsgröße, die zu diesem Zeitpunkt die höchste Behandlungsdringlichkeit besitzt, hier z.B. die Kontraktilität. Die Behandlungsdringlichkeit hängt ab von der Bewertung der Zustandsgrößen, ihrem aktuellen Trend und ihrer Gewichtung im Vergleich zu den anderen Zustandsgrößen. Die Berechnung erfolgt in der AES-2 Wissensbasis. Die drei Zustandsgrößen mit der höchsten Behandlungsdringlichkeit werden direkt neben der Empfehlung angezeigt. Das Piktogramm der Zustandsgröße, auf die sich die Empfehlung bezieht, ist als "aktiv" erkennbar. Eine farbige Pegelanzeige (einseitig) über den Piktogrammen gibt den Grad der Dringlichkeit an.

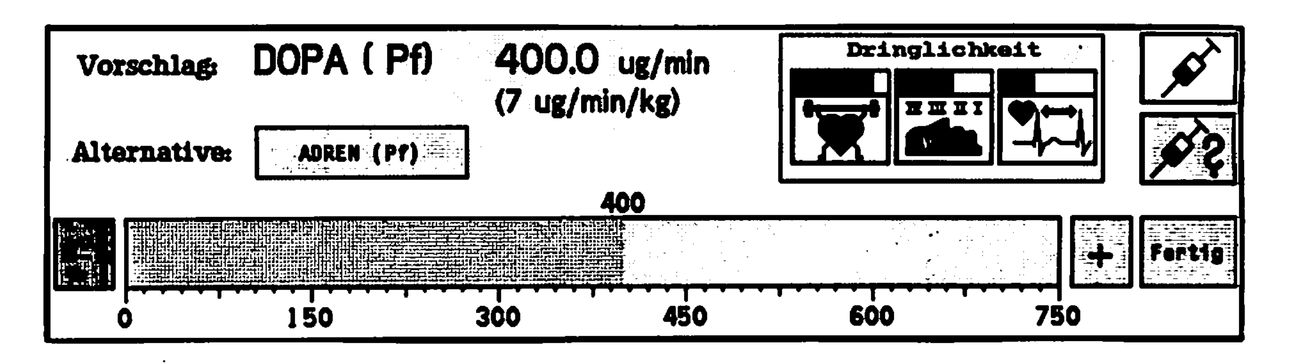


Bild 6.8: Therapieempfehlung zur Erhöhung der Kontraktilität.

Die dargestellte Empfehlung soll den Arzt in seiner Handlungsweise nicht festlegen; es stehen ihm verschiedene Interaktionsmöglichkeiten offen:

1. Der Arzt ist mit dem Vorschlag einverstanden: Er betätigt die Fertig-Taste und dokumentiert damit die Therapie für die Medikationsliste.
2. Der Arzt ist mit dem Medikament einverstanden, nicht aber mit der Dosis: Auf dem Werteschieber ist die vorgeschlagene Dosis in einem angemessenen Wertebereich bereits voreingestellt. Der Werteschieber nutzt die gesamte Breite des Bildschirms, so daß auch mit der Berühreingabe durch Verschieben leicht ein neuer Wert eingestellt werden kann. Der Wertebereich kann für außergewöhnlich hohe oder niedrige Dosierungen verändert werden. Ist der neue Wert eingestellt, verfährt der Arzt weiter wie in 1.
3. Der Arzt ist mit der Behandlung der Zustandsgröße einverstanden, will aber ein anderes Medikament nehmen. Die Wissensbasis verfügt für jede Zustandsgröße über mindestens einen alternativen Vorschlag. Diese Alternative ist unter dem vorgeschlagenen Medikament als Tastenbeschriftung angegeben. Bevorzugt der Arzt dieses Medikament (in Bild 6.8 Adrenalin), so berührt er diese Taste und erhält einen Dosierungsvorschlag für dieses Medikament. Die weitere Vorgehensweise ist wie 2 oder 1. Dem Arzt ist natürlich freigestellt, auch ein Medikament eigener Wahl zu verabreichen, das nicht in der Empfehlung berücksichtigt ist.

4. Der Arzt will eine andere Zustandsgröße behandeln: Die Piktogramme unter der Anzeige der Behandlungs-"Dringlichkeiten" können als Tasten benutzt werden, um den Therapievorschlag bezüglich einer anderen Zustandsgröße zu erhalten. Dieser Vorschlag erscheint dann ebenfalls mit einer Alternative auf der linken Seite. Weitere Vorgehensweise wie 3, 2 oder 1.

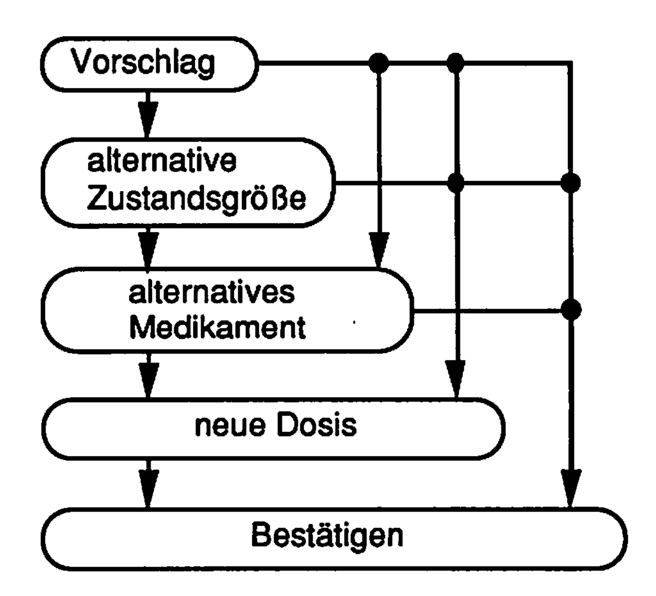


Bild 6.9: Interaktionsmöglichkeiten bei der Therapieempfehlung.

Die beschriebene Informationsgestaltung zeigt, wie auf einem Bildschirm diagnostische Informationen und mehrere Hypothesen zur Therapieplanung gleichzeitig angeboten werden können. Dadurch wird das Kurzzeitgedächtnis wesentlich entlastet, so daß mehr kognitives Potential für differenziertes Abwägen zur Verfügung steht. Die gesamte Information, die für eine neue Hypothesenbildung notwendig ist, kann auf dem Bildschirm dargestellt werden. Der Benutzer ist nicht externen Beobachter, sondern befindet sich durch die Interaktion innerhalb des Gesamtsystems ("in the loop"). Die Darbietung von Alternativen verhindert, daß dem System blindes Vertrauen geschenkt wird oder im Störungsfall nur langsam oder fehlerhaft reagiert wird, ein Problem, das besonders in hochautomatisierten Systemen auftritt (Kessel und Wickens 1982).

6.3.1.5 Medikationsliste (E)

In tabellarischer Form wird eine Liste der durchgeführten Therapiemaßnahmen mit Medikamentennamen, Menge und Zeit dargestellt. Zusätzlich ist angezeigt, ob ein Medikament über einen Perfusor verabreicht wurde. In dieser Liste kann mittels der beiden Pfeiltasten vor- oder zurückgeblättert werden. Der Zugriff auf diese Dokumentation kann für den Anästhesisten sowohl für die Diagnose als auch die Therapieplanung von Bedeutung sein.

Medikament	Menge	Einheit	Zeit	Perf.
KCL 7.45%	10	mval	9:45	
NL	100	gamma	9:48	
NL	50	gamma	10:08	Pf
XYLOCAIN	65	mg	10:12	
DIPIDOLOR	15	mg	10:15	
LASIX	10	mg	10:20	
DOPA	250	gamma	10:31	Pf

Bild 6.10: Medikationsliste.

6.3.1.6 Hinweise (F) und Kontrolle des Gesamtsystems (G)

Es ist geplant, Informationen über Routinetätigkeiten in die Wissensbasis zu integrieren, um den Anästhesisten an diese Tätigkeiten zu erinnern. So kann z.B. auf die regelmäßige Blutgasanalyse hingewiesen werden.

Zu jedem der angesprochenen physiologischen Teilsysteme ist eine Taste mit einem entsprechenden Piktogramm vorgesehen. Es wird die Taste desjenigen Teilsystems als "aktiviert" dargestellt, auf die sich der Bildschirminhalt bezieht. Diese Maßnahme erfüllt die Forderung, daß der Benutzer nie in einem Teilsystem isoliert sein darf (Moray 1981). Derzeit existiert zwar nur eine Wissensbasis für das kardiovaskuläre System, jedoch ist bereits in dieser Version der Bezug zum Gesamtsystem sichtbar gemacht worden.

6.3.2 Entwurf einer graphischen Erklärungskomponente

Im letzten Abschnitt wurde die Darstellung von Information über den Patientenzustand und die Empfehlung für die Therapieplanung erläutert. Im nächsten Schritt soll nun die wichtige Anforderung erfüllt werden, die Schlußfolgerungen des Expertensystems nachvollziehbar zu machen, d.h. der Benutzer soll die Möglichkeit haben, sich Klarheit über die Ursachen für eine bestimmte Bewertung oder Empfehlung des Systems zu verschaffen. Für die Darstellung einer Erklärungskomponente wurden bisher folgende Formen verwendet:

1. Der Text der für die Schlußfolgerung verwendeten Regeln wird dargestellt (Klocke 1987).
2. Ein Ausschnitt aus dem Entscheidungsbaum wird graphisch dargestellt, wobei jedem Zweig eine Regel entspricht.

Beide Formen erfordern sequentielles Lesen des verwendeten Regelwerkes der Wissensbasis. Im Routineeinsatz eines Systems ist für den Benutzer eine detailierte Betrachtung des Regelwerkes unangemessen. Ein schneller grafisch aufbereiteter Hinweis auf die maßgeblichen Informationen wäre hierfür besser geeignet.

Für den Aufbau der Erklärungskomponente des AES-2 ist die Tatsache entscheidend, daß die Darstellung aller notwendigen Informationen im Bildschirmentwurf bereits vorgesehen ist. Zur Erklärung einer Empfehlung müssen daher nur noch die Zusammenhänge zwischen bestimmten Informationen verdeutlicht werden. Diese Zusammenhänge können visuell durch farbige und/oder piktografische Markierung hergestellt werden. Die Erklärungskomponente des AES-2 wird in zwei Teile unterteilt:
1. die Begründung für die Bewertung einer Zustandsgröße,
2. die Begründung für die Therapieempfehlung.

6.3.2.1 Erklärungskomponente der Bewertung der Zustandsgrößen

Die Bewertung der Zustandsgrößen hängt hauptsächlich von der Bewertung physiologischer Meßgrößen ab. So lautet z.B. die Regel zur Bewertung der Kontraktilität:

Die Kontraktilität ist zu niedrig,
wenn der LAP normal oder über normal liegt
und der systolische Blutdruck unter normal ist.

Die Begründung für die Bewertung der Kontraktilität erscheint, wenn das Piktogramm der Zustandsgröße aktiviert wird. Daraufhin wird der Rahmen mit den Meßgrößen (B) zur Darstellung gebracht (falls er momentan nicht dargestellt ist) und es werden farbige piktografische Markierungen eingeblendet. In Bild 6.11 ist der Zusammenhang für die Bewertung "Kontraktilität zu niedrig" dargestellt.

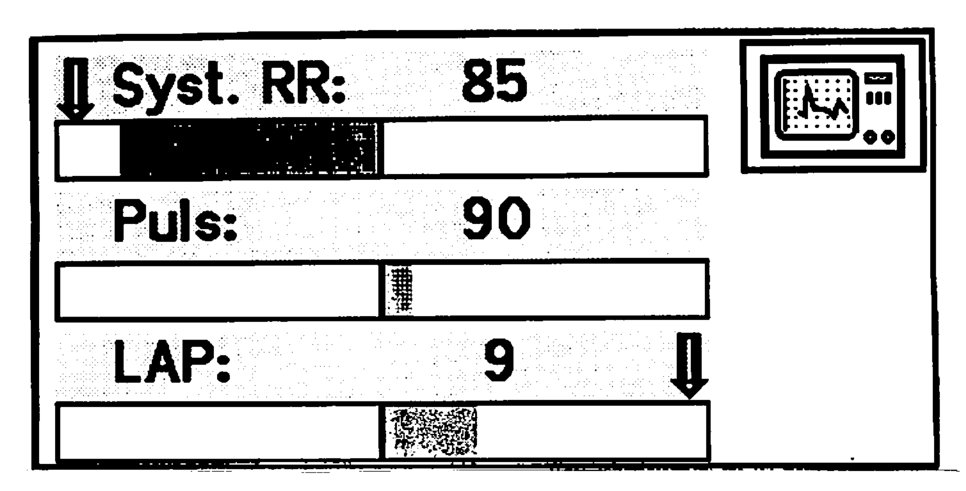


Bild 6.11: Meßgrößen mit Darstellung der Erklärungskomponente.

6.3.2.2 Erklärungskomponente der Therapieempfehlung

Die Therapieempfehlung im AES-2 ist immer auf genau eine Zustandsgröße bezogen. Der Bezug zu dieser Zustandsgröße ist permanent durch das aktive Piktogramm im Feld "Dringlichkeit" sichtbar und muß daher nicht zusätzlich als Erklärung angegeben werden. Darüber hinaus ist die Auswahl und Dosis des empfohlenen Medikaments abhängig von Patientendaten und dem bisherigen Therapieplan. Dieser Teil einer Erklärung des Therapievorschlags kann durch Betätigung der Piktogrammtaste mit dem Fragezeichen innerhalb des Therapierahmens (D) angefordert werden. Daraufhin werden Markierungen in den Patientendaten (C) oder in der Medikationsliste (F) sichtbar. Als Beispiel soll die Erklärung für die Therapieempfehlung "400 gamma Dopamin" erläutert werden. Es wird zunächst der Eintrag des Gewichts in den Patientendaten markiert (Farbbild 6.1, Anhang B-3), da die Dosis vom Körpergewicht abhängig ist. Außerdem ist der bisherige Therapieplan in diesem Vorschlag berücksichtigt: da bereits zuvor 250 gamma Dopamin verabreicht wurden, die nicht zur Normalisierung der Kontraktilität führten, wird eine höhere Dosis vorgeschlagen. Der Eintrag dieser Maßnahme wird deshalb in der Medikationsliste farbig markiert (Farbbild 6.1).

Anhand der Gestaltung der Benutzerschnittstelle des Anästhesie Entscheidungs-Unterstützungssystems AES-2 konnte gezeigt werden, wie die zuvor entwickelten Methoden und Werkzeuge zu einer effektiven Vorgehensweise bei der Gestaltung eingesetzt werden können. Ausgehend von kognitiven Aspekten des Entscheidungsprozesses bei der Überwachung und Steuerung des Patientenzustands wurde eine Visualisierung der relevanten Informationen für jede Phase des Prozesses realisiert. Es wurde eine hierarchische Strukturierung der Schnittstelle vorgenommen, die sich an den verschiedenen Aspekten der Aufgabe des Anästhesisten orientiert. Innerhalb der einzelnen Teilschnittstellen konnte durch Verwendung von Farb- und Formkodierung sowie von Piktogrammen die für Diagnose und Therapie notwendige Information dargestellt werden.

7. Zusammenfassung

Die vorliegende Arbeit stellt eine Software-Entwicklungsumgebung vor und zeigt einen Lösungsweg für die Entwicklung von farbgraphischen, multimodalen Benutzerschnittstellen, die zur Unterstützung des Arztes bei der Aufgabe der Patientenüberwachung eingesetzt werden können. Die Berücksichtigung der besonderen Merkmale des medizinischen Bereichs erfordert ein iteratives Vorgehen, um die bestmögliche Anpassung des Systems an die Anforderungen des Benutzers, an seine perzeptiven und kognitiven Fähigkeiten und Grenzen sowie an seine Arbeitsumgebung zu erreichen.

Als geeignete Vorgehensweise wurde das evolutionäre Prototyping vorgestellt; hierbei wird der Benutzer von Projektbeginn an in den Entwicklungsprozeß miteinbezogen, so daß im Projektmanagement in den Bereichen der Planung und Durchführung andere Voraussetzungen berücksichtigt werden müssen als in Projekten mit konventioneller Vorgehensweise. Evolutionäres Prototyping kann besonders effektiv durchgeführt werden, wenn dem Systementwickler eine leistungsfähige Software-Entwicklungsumgebung zur Verfügung steht.

Die in dieser Arbeit konzipierte Entwicklungsumgebung vereinfacht den Entwurf prototypischer Versionen und bietet gleichzeitig Ansätze zur heuristischen und objektiven Bewertung. Den Kern der Entwicklungsumgebung bildet eine objektorientierte Bibliothek mit wiederverwendbaren Bausteinen für die Interface-Gestaltung. Diese Bausteine erfüllen die Anforderungen eines sog. "user interface management system (UIMS)"; im Unterschied zu den bekannten UIMSs wurden Schwerpunkte auf die hierarchische Strukturierung einer Benutzerschnittstelle sowie die direkt-manipulative, multimodale Interaktion gelegt. Der akustische Kanal wird sowohl bei der Eingabe (Spracherkennung) als auch bei der Ausgabe (akustische Meldungen) miteinbezogen. Die Implementierung einer Benutzerschnittstelle unter Verwendung der Bausteine wurde anhand einer Infusionspumpen-Kontrolleinheit beschrieben. Dabei wurde ausgehend von einer hierarchischen Analyse die Benutzeraufgabe in die Syntax und Semantik einer die Schnittstelle beschreibenden Kodierung (DIC) umgesetzt.

Aufbauend auf die Baustein-Bibliothek enthält die Entwicklungsumgebung verschiedene Editoren:

- Ein Entwurfseditor unterstützt das interaktive Erstellen einer Benutzerschnittstelle. Die interaktiven Objekte des Interface-Baukastens können ähnlich wie bei Editoren im CAD-Bereich positioniert und modifiziert werden. Es entsteht dabei jedoch nicht nur ein "Bild" der Schnittstelle sondern sofort das interaktive System selbst. Der

Editor stellt die Möglichkeit bereit, für die interaktiv entworfene Schnittstelle den entsprechenden kompilierbaren Quellkode (DIC-Syntax) zu erzeugen. Auch bereits kompilierte Programme von Benutzerschnittstellen können zur Laufzeit durch den Editor von einem separaten Bildschirm innerhalb desselben Rechnernetzwerkes aus modifiziert werden.

- Ein Farbeditor (CEDI) ermöglicht die interaktive Manipulation von Farben in einem von 3 Farbenräumen, die den drei Verarbeitungsstufen des visuellen Systems entsprechen. Der Simultankontrast zweier Farben kann in verschiedenen Infeld/-Umfeld Situationen beurteilt werden. Die Zuordnung von Farbkoordinaten zu einem Bezeichner kann in einer Farbdatenbank abgespeichert werden, wodurch diese Farben dann einheitlich von jeder Applikation mit Hilfe des entsprechenden Bezeichners aufgerufen werden können. Die Übertragbarkeit von Farben wird durch einen genormten Farbenraum (CIELAB) ermöglicht. Der farbliche Feinabgleich aller Farben innerhalb einer Applikation kann zur Laufzeit von einem zweiten Bildschirm aus vorgenommen werden, der über das Rechnernetzwerk angekoppelt wird.
- Die Interaktion durch Spracheingabe wird durch ein Werkzeug für Wortschatzentwurf, -training und -test unterstützt. Der Einsatz von Sprachkommandos in einer Applikation wurde durch eine Einbettung in das Client-Server Modell erreicht. Die Auswertung von multimodalen Eingaben ist im Interface-Baukasten auf der Ebene der einzelnen Objekte realisiert.
- Akustische Meldungen zur Informationsausgabe werden zunächst mit einem kommerziellen Werkzeug generiert und editiert. Es wurde für die so erstellten akustischen Meldungen eine Bibliothek mit interaktiver Benutzerschnittstelle eingerichtet, die eine komfortable Verwaltung dieser Meldungen und den Zugriff durch spezielle Präsentations-Objekte des Interface-Baukastens ermöglicht.

In dieser Entwicklungsumgebung wird nicht nur der Entwurf einer Benutzerschnittstelle unterstützt; die Möglichkeit der Bewertung ist für verschiedene Aspekte der Gestaltung folgendermaßen berücksichtigt worden: Der Entwurfseditor unterstützt die Berechnung der geometrischen Komplexität eines Entwurfs. Die Bewertung von Piktogrammen kann mit Hilfe eines Fragebogens durchgeführt werden, der für eine Beurteilung durch die Benutzergruppe entworfen wurde. Die Berechnung der Buntheit (Mittelwert und Standardabweichung) konnte die Bewertung farbiger Bildschirmdarstellungen durch quantitative Kenngrößen ergänzen. Es wurde eine Korrelation zwischen den Buntheitswerten eines farbigen Bildschirmlayouts und dem durch den Benutzer eingeschätzten subjektivem Urteil festgestellt. Die Bewertung eines Wortschatzes zur Spracherkennung kann anhand eines Testbetriebes der Worterkennung vorgenommen werden. Für die Bewertung von akustischen Meldungen wurde wiederum ein Fragebogen entworfen, der eine rechnergestützte Beurteilung durch den

Benutzer ermöglicht.

Mit Hilfe dieser Entwicklungsumgebung kann der Prozeß des evolutionären Prototypings für die genannten Gestaltungsaufgaben effizienter durchgeführt werden, d.h. die Anzahl der notwendigen Iterationen und die Zeit für jede Iteration wird verringert. Am Beispiel der Gestaltung eines Anästhesie Entscheidungs-Unterstützungssystems wurde ein systemergonomischer Ansatz zur Gestaltung eines komplexen Informationssystems ausgehend von kognitiven Aspekten der Aufgabe durchgeführt. Insbesondere wurden die Strukturierung und Visualisierung von eine Entscheidung unterstützender Information durch graphische Mittel sowie eine graphische Erklärungskomponente behandelt. Es wurde dabei aufgezeigt, wie die perzeptiven und kognitiven Fähigkeiten und Grenzen des Menschen im Entwurf der Benutzerschnittstelle durch Einsatz der beschriebenen Methoden und Werkzeuge systematisch berücksichtigt werden können.

Quellenverzeichnis

Anderson, S. A. (1981) Cognitive skills and their acquisition. Hillsdale (N.J.): Erlbaum Associates

Apple (1988) Inside Macintosh. Volume 1-5, Apple Computer

Apple (1988) HyperCard Benutzerhandbuch. Apple Computer

Arend, U.; Wandmacher, J. (1988) Gestaltungsprinzipien für Piktogramme und ihr Einfluß auf die Menüauswahl. Kurzfassung zum Vortrag auf dem Workshop "Werkzeuge zur Gestaltung von Benutzerschnittstellen, Darmstadt

Bailey, R. W. (1982) Human Performance Engineering. NJ: Prentice Hall

Bernotat, R.; Hunt, D. (ed.) (1977) Curricula on Ergonomics, Meckenheim Forschungsinstitut für Anthropotechnik

Bersoff, E. H.; Henderson, V. D.; Siegel, S. G. (1980) Software Configuration Management. NJ: Prentice Hall

Bertin, J. (1982) Graphische Darstellungen und die graphische Weiterverarbeitung der Information. Berlin: de Gruyter

Boehm, B.W. (1976) Software Engineering, in: IEEE Transactions on Computing 25, No. 12, 1226-1241

Boehm, B.W.; Graz, T.E.; Seewald, T. (1984) Prototyping versus Spezifying: A Multiproject Experiment. IEEE Trans. SE 10, no.3, 290-303

Bonsiepe, G. (1968) A method of quantifying order in typographic design. Journal of Typographic Research, 2, 203-220

Broadbent, D. (1971) Decision and stress. New York: Academic Press

Budde, R.; Kuhlenkamp, K. (1984) Approaches to Prototyping. Springer Verlag

Card, S.; Moran, T. & Newell, A. (1983) The psychology of human computer interaction.Hillsdale, NJ: Erlbaum Associates

Case, A. (1985) Information Systems Development: Principles of Computer-Aided Software Engineering, NJ: Prentice-Hall

Chase, W. G.; Ericsson, A. (1981) Skilled memory, in: Anderson, S. A. Cognitive skills and their acquisition, Hillsdale (N.J.): Erlbaum Associates

Chase, W. G. (1986) Visual Information Processing. In: Handbook of Perception and Human Performance, Vol. II chapter 28, New York: John Wiley and Sons

Christ, R.E. (1975) Review and analysis of color coding research for visual displays. Human Factors, 17, 542-570

CIE Commission international de l'eclairage (1978) Recommendations on uniform color space, colordifference equations, and psychometric color terms, Supplement No. 2 to CIE No. 15

Cowan, W.B. (1983) An inexpensive scheme for calibration of a color monitor in terms of CIE standard coordinates. Computer Graphics, 17, 315-321

Cox, B. (1986) Object-Oriented Programming - An evolutionary approach. Addison-Wesley

Craik, F.; Lockhart, R. (1972) Levels of processing: A framework for memory research. Journal of Verbal Learning and Verbal Behaviour 11, 671-684

Darwin, C.; Turvey, M.; Crowder, R. (1972) An analog of the Sperling partial report procedure. Cognitive Psychology 3, 255-267

Davidoff, J. (1987) The Role of Colour in Visual Displays. Intern. Reviews of Ergonomics, 1, 21-42; Oborne, D.J.(Ed.), Taylor and Francis

Dick, W. und Carey, L. (1978) The systematic design of instruction. Glenview(IL): Scott, Foresman

DIN 5033 Teil 2 Farbmessung: Normvalenz-Systeme, Beuth Verlag

DIN 5033 Teil 3 Farbmessung: Farbmaßzahlen, Beuth Verlag

DIN 6164 Teil 1 System der DIN-Farbenkarte für 2° Normalbeobachter, Beuth Verlag

DIN 66234 (1980) Teil 8: Gestaltung von Bildschirmarbeitsplätzen, Beuth Verlag

Döring, B. (1986) Systemergonomie bei komplexen Systemen. In: Arbeitsorganisation und Neue Technologien (R, Hackstein, F.-J. Heeg, F. v. Below, Hrsg.), Berlin Springer Verlag

Draper, S. and Norman, D. (1984) Software Engineering for User Interfaces. IEEE Transactions, 214-220

Dzida, W. (1983) Das IFIP-Modell für Benutzerschnittstellen. Office Management, Sonderheft, 6-8

Easterby, R. (1984) Tasks, processes and display design. In: R. Easterby, H. Zwaga, Information Design, John Wiley, 19-36

Endres, A. (1988) Software-Wiederverwendung: Ziele, Wege und Erfahrungen. Informatik-Spektrum 11, 83-95

Eriksen, C. W.; Hake, H. N. (1955) Absolute judgement as a function of stimulus range and number of stimulus and response categories. J. of Experimental Psychology 49, 323-332

Fairley, R. E. (1985) Software Engineering Concepts. MacGraw-Hill

Farooq, M.U.; Dominick, W.D. (1988) A Survey of Formal Tools and Models for Developing User Interfaces. Int. J. Man-Machine Studies 29, 479-496

Fittkau, B.; Müller-Wolf, H.; Schulz von Thun, F. (1980) Kommunizieren lernen (und umlernen). Braunschweig: Westermann

Floyd, Ch. (1981) A Process-oriented Approach to Software Development, in: Systems Architecture, Proc. of the 6th European ACM Regional Conference, Westbury House, 285-294

Floyd, Ch. (1984) A Systematic Look at Prototyping. In: R. Budde; K. Kuhlenkamp: Approaches to Prototyping, Springer Verlag

Floyd, Ch.; Keil, R. (1984) Integrative Systementwicklung. BMFT Forschungsbericht DV 84-003

Floyd , Ch. (1987) STEPS - Eine Orientierung der Softwaretechnik auf sozialverträgliche Technikgestaltung. In W. Schönpflug, M. Wittstock: Software-Ergonomie 87, Teubner-Verlag, 500-503.

Foley, D.J.; van Dam, A. (1982) Fundamentals of Interactive Computer Graphics. Addison-Wesley Publishing

Foley, J.; Wallace, V; Chan, P. (1984} The Human Factors of Computer Graphics Interaction Techniques. IEEE Computer Graphics & Applications, November 1984, 13-48

Foley, J. D. (1987) Interfaces for Advanced Computing. Scientific American, Oct.1987, 127-135

Fukuzumi, S., and Hayashi, Y. (1989) Luminance and Stimulus Purity of VDT Display Color in Terms of Readability. Int. Journal of Human-Computer Interaction, Vol.1, Nr.2., Ablex Publishing

Gettys, J.; Newman, R.; Scheifler, R. (1988) Xlib-C Language X Interface. MIT, MA

Gravenstein, J. (1982) As for pilots, instruments important. Eng. Med. Biol. 1 (1), 23-25

Green, M. (1984) Report on Dialogue Specification Tools. Computer Graphics Forum 3, 305-313

Green, M. (1985) The University of Alberta User Interface Management System. ACM Siggraph Vol. 19, No. 3, 205-213

Grimm; Haller; Sybre; Rudolf. (1983) Bildschirme in der Prozeßleitwarte. Verlag TÜV Rheinland

Grudin, J.; Ehrlich, S.; Shriner, R. (1987) Positioning Human Factors in the User Interface Development Chain. ACM CHI+GI, 125-131

Hard, A. (1969) Qualitative attributes of colour perception. Tagungsbericht Internationale Farbtagung "Color 69" Stockholm 1969, Göttingen: Muster-Schmidt, 351-368

Harker, S. (1987) Rapid Prototyping as a Tool for User Centered Design. in G. Salvendy: Cognitive Engineering in the Design of Human-Computer Ineraction and Expert Systems, Elsevier, 365-372

Häusing, M. (1975) Gestaltungsrichtlinien für Bildschirmarbeitsplätze mit Rastersichtgeräten, Werthoven: FAT-Bericht Nr. 26

Haux, R. (1987) Expertensysteme in der Medizin. Habilitationsschrift RWTH Aachen

Hayes, P.; Szekely, P.; Lerner, R. (1985) Design Alternatives for User Interface Management Systems based on Experience with COUSIN. CHI'85 Proceedings

Heinz, A. (1989) Graphische Gestaltung der Benutzeroberfläche eines Entscheidungs-unterstützungs-Systems in der Anästhesiologie. Diplomarbeit am Helmholtz-Institut, RWTH Aachen

Henninger, Kaminsky, Schmale. (1980) Ergonomie am Arbeitsplatz. Friedrich Kiehl Verlag

Hering, E. (1920) Grundzüge der Lehre vom Lichtsinn. Springer-Verlag

Hick, W. (1952) On the rate of gain of information. Quaterly journal of Experimental Psychology, 4, 11-26

Hoffmann, G. (1987) Organisation und Programmierung einer Dialogebene für die Kommunikation zwischen Sprecher, Spracherkennungssystem und Hostrechner. Diplomarbeit Helmholtz-Institut, RWTH Aachen

Hofstetter, H. (1985). Psychologische Probleme und Verfahren bei der Softwareentwicklung, in: SWT 5, Heft 2, 63-72

HP (1989), HP-Widget User Manual

Hübner, W.; Lux-Mülders, G.; Muth, M. (1987) THESEUS. Die Benutzeroberfläche der UNIBASE-Softwareentwicklungsumgebung. Springer-Verlag

Hutchin, E.; Hollan, J. Norman, D (1986) Direct Manipulation Interfaces, in: D. Norman und S. Draper, User Centered System Design, Lawrence Erlbaum, 87-124

Jacob, R.; Egeth, H.; Bevon, W. (1976) The face as a data display. Human factors 18, 189-200

Jameson, D. ; Hurvich, L. M. (1972) Visual Psychophysics. Handbook of Sensory Physiology Band VII/4 , Springer-Verlag

Kaster, J.; Kraiss, K. F.; Küttelwesch, H. (1985) Ein Verfahren zur grafisch interaktiven Definition von Farbvalenzen, in: Software Ergonomie '85, H. J Bullinger (Ed.)

Kessel, C. J.; Wickens, C. D. (1982) The transfer of failure-detection skills between monitoring and controlling dynamic systems. Human Factors 24, 49-60

Klocke, H. (1985) Entwicklung einer Mensch-Rechner Schnittstelle für ein Anästhesie-Informationssystem unter Berücksichtigung ergonomischer Gesichtspunkte. Angewandte Informatik, No. 5, 197-208

Klocke, H. (1987) Konzeptentwicklung und Realisierung eines Informationssystems zur Dokumentation, Überwachung und Entscheidungsunterstützung während der Anästhesie. Dissertation, RWTH Aachen

Kopala, C. (1979) The use of color-coded symbols in a highly dense situation display, in: Bensel, C. Proceedings of the 23rd annual meeting of the Human Faktors Society, Santa Monica

Kraiss, K.-F. (1985) Fahrzeug- und Prozeßführung, kognitives Verhalten des Menschen und Entscheidungshilfen, Springer-Verlag

Langen, M.; Rau, G. (1990) A development environment for the design of multimodal, colourgraphic human-computer interfaces, in: D. Diaper (Ed.) Human-Computer Interaction INTERACT 90, Elsevier North Holland.

Lang, H. (1978) Farbmetrik und Farbfernsehen. München: Oldenbourg

Lees, J.; Farman, M. An investigation of the design and performance of traffic control devices. Visible Language, 4, 7-38

Lodding, K. N. (1983) Iconic Interfacing. Computer Graphics and Application, March 1983, 11-20

Löwgren, J. (1988) History, State and Future of User Interface Management Systems. SIGCHI Bulletin, Vol. 20, No. 1, July 1988, 32-44

McCauley, M.E. (1984) Human Factors In Voice Technology. in: F. Muckler: Human Factors Review: 1984. Human Factors Society, Santa Monica CA, 131-166

Meister, D., Sullivan, D. J. (1969) Guide to Human Engineering Design for Visual Displays. Engineering Psychology Branche, Office of Naval Research, Washington D.C.

Meister, D. (1986) Human Factors Testing and Evaluation, Elsevier, Amsterdam

Miller, G. A. (1956) The magical number seven plus or minus two: Some limits on our capacity for processing information, Psychological Review 63, 81-97

Mishkin, M.; Appenzeller, T. (1987) Die Anatomie des Gedächtnisses. Spektrum der Wissenschaft, August 1987, 94-104

Moray, N. (1981) The role of attention in the detection of errors and the diagnosis of failures in man-machine systems, in: Rasmussen, J.; Rouse, W. B.: Human detection and diagnoses of system failures, New York: Plenum Press

Moses, F. L. ; Maisano, R. E. ; Bersh, P. (1979) Natural associations between symbols and military information in: Bensel, C. Proceedings of the 23rd annual meeting of the Human Faktors Society, Santa Monica

Mumford, E.; Welter, G. (1984) Benutzerbeteiligung bei der Entwicklung von Computersystemen. Erich Schmidt Verlag, Berlin.

Murch, G. M. (1984) Physiological Principles for the Effective Use of Color. IEEE Computer Graphics and Applications, November, 49-54

Murchner, B.; Oppermann, R.; Paetau, M.; Pieper, M.; Simm, H. & Stellmacher, I. (1987) EVADIS - Ein Leitfaden zur softwareergonomischen Evaluation von Dialogschnittstellen. In: W. Schönpflug, M. Wittstock, Software-Ergonomie '87, Teubner Verlag, 307-316

Myers, B. A. (1984) The User Interface for Sapphire. Computer Graphics and Application, Dec. 1984, 13-23

Myers, B. (1987) Creating Interaction Techniques by Demonstration. IEEE Computer Graphics and Applications, September

Myers, B. (1987b) Gaining General Acceptance for UIMS. Computer Graphics 21:2
Nachrichtentechnische Gesellschaft (1986) Software-Qualitätssicherung. NTG-Schrift 12-51, VDE-Verlag

Nelson, R. (1981) The design of advertising. Brown Company Publishers

Neuser, R. (1987). Nutzergerechte Ausgestaltung von Software durch Prototyping. Diplomarbeit Inst. f. Arbeitswissenschaft, RWTH Aachen

Newhall, S.M. ; Nickerson, D. ; Judd, D.B. (1943) Final report of the O.S.A. subcommittee on the spacing of the Munsell colors. J. opt. Soc. Amer. 33, 385-418

Nolte, E. (1990) Entwurf und Implementierung eines graphischen Editors zum Aufbau interaktiver Benutzerschnittstellen. Diplomarbeit am Helmholtz-Institut, RWTH Aachen

Norman, D.; Bobrow, D. (1975) On data-limited and resource-limited processing. Journal of Cognitive Psychology 7, 44-60

Norman, D.; Draper, S.; Bannon, L. (1986) Glossery In: D. Norman, S. Draper (Ed.) User Centered System Design. Hillsdale (NJ) Lawrence Erlbaum Ass.

Norman, D. (1986) New Views of Information Processing: Implications for Intelligent Decision Support Systems. In: Hollnagel, E.; Mancini, G.; Woods, D. D., Intelligent Decision Support in Process Enviroments. Springer-Verlag

Nye, A. (1988), Xlib Programming Manual. O'Reilly & Associates Inc.

Olsen, D. Dempsey, E. (1983) SYNGRAPH: A Graphic User Interface Generator. Computer Graphics 17:3

Oppermann, R. (1983). Forschungsstand und Perspektiven partizipativer Systemgestaltung. Bonn-St. Augustin: Bericht Nr.140 der Gesellschaft für Mathematik und Datenverarbeitung (GMD)

Parnas, D. (1972) On the Criteria to be Used in Decomposing Systems into Modules. Comm. ACM 15, No. 12, 1053-1058

Perlman, G. (1988) Software Tools for User Interface Development. in: M. Helander (Ed.), Handbook of Human-Computer Interaction. Elsevier North Holland, 819-833

Pfaff, G. (1985) User Interface Management Systems. Springer-Verlag

Pfendler, C.; Widdel, H., (1986) Vigilance performance when using color on electronic displays. Perceptual and motor skills 63, 939-944

Platz, G. (1983) Methoden der Softwareentwicklung. München: Hanser Verlag

Pomeranz, J. und Kubovy, M. (1986) Theoretical Approaches to Perceptual Organization. in: K. Boff, Handbook of Perception and Human Performance, Vol. 2, John Wiley & Sons, chapter 36

Posner, M.; Nissen, J.; Klein, R. (1976) Visual dominance: An information processing account of its origins and significance. Psychological Review 83, 157-171

Pressman, R. (1987) Software Engineering - A Practitioner's Approach. MacGraw-Hill

Rasmussen, J. (1976) Outlines of a hybrid model of the process plant operator. In: Sheridan, T. B.; Johannsen, G. (eds.), Monitoring Behaviour and Supervisory Control, Plenum Press, New York

Rasmussen, J. (1986) Information Processing and Human-Machine Interaction. Elsevier, North Holland

Rau, G. (1979): Ergonomische Überlegungen bei der Gestaltung komplexer medizinischer Instrumentierung unter Einsatz von Mikroprozessoren, Biomedizinische Technik Band 24, Ergänzungsband

Rau, G.; Trispel, S. (1982) Ergonomic Design Aspects in Interaction between Man and Technical Systems in Medicine. Medical Progress through Technology 9: 153-159

Richter, K. (1979) Beschreibung von Problemen der höheren Farbmetrik mit Hilfe des Gegenfarbensystems Forschungsbericht 61, BAM Berlin

Rowe, :L.; Shoens, K. (1983): Programming language constructs for screen definition. IEEE Transactions on Software Engineering, 9, 31-39

Schecke, T.; Rau, G.; Klocke, H.; Käsmacher, H.; Hatzky, U.; Kalff, G.; Zimmermann, H. (1988) Knowledge-Based Decision Support in Anesthesia: A Case Study. Proceedings of the 1988 IEEE International Conference on Systems, Man and Cybernetics, 962-965

Scheifler, G.W.; Gettys, J. (1986) The X-Window System. ACM Transactions on Graphics, Vol. 5, No. 2, April

Schmidt, R. F.; Thews, G. (1976) Physiologie des Menschen. 17.Aufl. Springer-Verlag

Schröter, A. (1989) Integration von multimodaler Interaktion in einen objekt-orientierten Baukasten interaktiver Schnittstellenelemente. Diplomarbeit am Helmholtz-Institut, RWTH Aachen

Schulz, A. (1988) Software-Entwurf. Oldenbourg-Verlag

Sheridan, T. B. (1981) Understanding human eror and aiding human diagnostic behavior in nuclear power plants. In: Rasmussen, J.; Rouse, W. B.:Human detection and diagnoses of system failures, New York: Plenum Press

Shneiderman, B. (1982), The future of interactive systems and the emergence of direct manipulation, Behaviour & Information Technologie (GB), Vol. 1, no.3, 237-256

Siebert, J.L.; Hurley, W.D.; Bleser, T.W. (1988) Design and Implementation of an Object-oriented User Interface Management System. in: R. Hartson, D. Hix (Eds.): Advances in Human-Computer Interaction - Volume 2. Ablex Publishing, 175-213

Sickert, K. (1984) Automatische Spracheingabe und Sprachausgabe. Verlag Markt und Technik

Simpson, C.A.; McCauley, M.E.; Roland, E.F.; Ruth, J.C.; Williges, B.H. (1987) Speech Controls And Displays. In: G. Salvendy: Handbook of Human Factors, 1490-1525 John Wiley & Sons

Smith, W., (1987) Ergonomic Vision. In: H.J. Durett (Ed.): Color and the Computer. Academic Press, Inc.

Sneed, H. (1980) Software-Entwicklungsmethodik. Verlagsgesellschaft Rudolf Müller

Spitta, T. (1989) Software Engineering und Prototyping. Springer Verlag

Stanford, L; McInteyre, J.; Nelson, T. and Hogan, J. (1988) Affective responses to commercial and experimental auditory alarm signals for anesthesia delivery and physiological monitoring equipment. Int. J. of Clinical Monitoring and Computing 5, 111-118

Staufer, M. J. (1987) Piktogramme für Computer. De Gruyter

Streveler, D.J.; Wasserman, A.I. (1985) Quantitative Measures of the spatial properties of screen designs. In: B. Shackel (ed.), Human-Computer Interaction - INTERACT 84. Elsevier North Holland

ten Hagen, P.; Derksen, J. (1985) Parallel Input and Feedback in Dialogue Cells. In: G. Pfaff: User Interface Management Systems. Springer, 109-124

Thompson, T. (1989) The Next Step. Byte, March 1989, 265-269

Treisman, A. (1960) Contextual cues in selective listening. Quaterly Journal of Experimental Psychology 12, 242-248

Treisman, A. (1986) Properties, Parts and Objects. In: Handbook of Perception and Human Performance, Vol. II chapter 36, John Wiley

Tullis, T. S., (1981) An evaluation of alphanumeric, graphic and color information displays. Human Factors, 23, 541-550

Tullis, T. S. (1983) The formatting of alphanumeric displays: a review and analysis. Human Factors, 25(6), 657-682

Tullis, T. S. (1988) A System for Evaluating Screen Formats: Research and Application, in: H. R. Hartson und D. Hix (eds.), Advances in Human-Computer Interaction - Volume 2. Ablex Publishing, 214-286

von Helmholtz, H. (1896) Handbuch der physiologischen Optik. 2. Aufl. Hamburg Leipzig: L.Voss

van Nes, F. L. (1985) On the use of layout, colour and typography on visual display terminals, IPO annual progress report 20, 110-119

VIP (1988) VAPS - Virtual Prototyping System. Quebec: Virtual Prototype Inc.

v. Kries, J. (1905) Theoretische Studie über die Umstimmung des Sehorgans. Festschrift der Universität Freiburg

Ware, C; Cowan, W. (1982) Changes in perceived color due to chromatic interactions. Vision Research Vol. 22, 1353-1362

Wertheimer, M. (1923) Untersuchungen zur Lehre von der Gestalt. II. Psychologische Forschung, 4, 301-350

Wetmore, R. (1988) MacApp 2.0 ViewEdit User's Guide. Apple Computer

Wickens, C.; Kessel, C. (1980) The processing resource demands of failure detection in dynamic systems. J. of Experimantal Psychology: Human Perception and Performance 6, 564-577

Wickens, C. D. (1984) Engineering Psychology and Human Performance. Ch. E. Columbus (Ohio): Merrill Publisching Co.

Widdel, H, (1987), Semantisches Differential zur subjektiven Bewertung farbiger, digitaler Landkarten. Werthoven: FAT

Williges, R.C.; Williges, B.H.; Elkerton, J. (1987) Software Interface Design. In: G. Salvendy: Handbook of Human Factors. John Wiley and Sons

Wilson, J.; Rosenberg, D. (1988) Rapid Prototyping for User Interface Design. In: M. Helander (Ed.), Handbook of Human-Computer Interaction. Elsevier North Holland, 859-87.

Wippich, W. (1984) Lehrbuch der angewandten Gedächtnispsychologie. Band 1. Stuttgart-Berlin-Köln-Mainz Kohlhammer

Woods, D.; Wise, J.; Hanes, L. (1981) An evaluation of nuclear power plant safety parameter display systems, in: Sugarman,R. C. Proceedings, 25th annual meeting of the Human Factors society, Santa Monica, CA: Human Factors

Woods, D. D. (1986) Paradigms for Intelligent Decision Support. In: Hollnagel, E.; Mancini, G.; Woods, D. D.: Intelligent Decision Support in Process Enviroments, Springer-Verlag

Wyszecki, G. (1986) Color Appearance in: Handbook of Perception and Human Performance, Boff, K. (Ed.), New York: John Wiley & Sons, chpt. 9

Zwanga, H. J.; Boersema, T. (1983) Evaluation of a set of graphic symbols. Applied Ergonomics, 14, 43-54

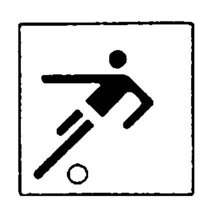
Anhang A: Studie zur Bewertung von Piktogrammen

A.1 Aufbau des Fragebogens

Bewertung von Piktogrammen im medizinischen Bereich

- FRAGEBOGEN -

Piktogramme sind bildhafte Darstellungen von Begriffen oder Sachverhalten. Breite Verwendung finden Piktogramme als Wegweisersystem von Flughafen oder Bahnhof sowie zur Kennzeichnung von Sportarten; aber auch im medizinischen Bereich werden Piktogramme zunehmend verwendet.


Folgende Vorteile sind dadurch möglich:
- Erkennen eines Begriffs "auf einen Blick"
- Allgemeinverständlichkeit (d.h. unabhängig von Landessprache oder Fachfremdworten)
- benötigen weniger Platz

Diese Vorteile können jedoch nur erreicht werden, wenn der Entwurf von Piktogrammen sorgfältig durchgeführt wird. Eine wichtige Phase in diesem Entwurfsvorgang ist die Bewertung von Piktogrammen durch einen ausgewählten Benutzerkreis.

Teilnehmer: GESCHLECHT: m / w

MUTTERSPRACHE deutsch / sonstige

ALTER:
bis 30 / 31-40 / 41-50 / 51-60 / über 60 Jahre

BERURSGRUPPE:
Pflegepersonal / Ärzteschaft / Student / sonstige

ARBEIT IN DER KARDIOCHIRURGIE:
bislang gar nicht / unter 1 Jahr / länger als 1 Jahr

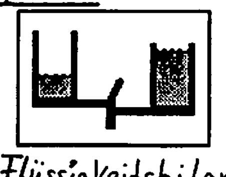
Bevor der Fragenteil zur Bewertung beginnt, sollen Sie Ihre Einstellung zur Verwendung von Piktogrammen im medizinischen Bereich einschätzen. Dazu können Sie eine beliebige Ziffer auf der Skala zwischen "-3" und "+3" ankreuzen, abhängig davon, ob eher der links oder der rechts stehende Begriff zutrifft.

1. ICH HALTE PIKTOGRAMME FÜR: nicht sinnvoll sinnvoll
-3 -2 -1 0 +1 +2 +3

TEIL I

Auf den folgenden Seiten werden Ihnen Piktogramme vorgestellt, die z.B. auf dem Bildschirm eines Gerätes zur Patientenüberwachung erscheinen könnten. Sie sollen Begriffe bezüglich des Patientenzustands oder bestimmte Informationsgruppen widerspiegeln.
Unter jeder Abbildung ist eine freie Zeile, auf die Sie einen Begriff schreiben können, der Ihrer Meinung nach den Inhalt des Piktogramms am besten ausdrückt. Außerdem können Sie bewerten, inwieweit sie die Darstellung sympathisch und erkennbar finden.

Beispiel:

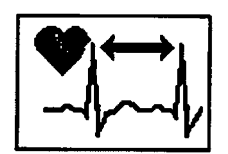


Flüssigkeitsbilanz

Bild: schlecht erkennbar -3 -2 -1 0 (+1) +2 +3 gut erkennbar

Ästhetik: unsympathisch -3 (-2) -1 0 +1 +2 +3 sympathisch

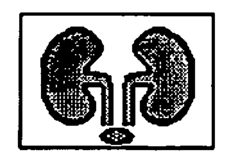
Sollte ein Begriff ausnahmsweise nicht ausreichen, können Sie auch eine kurze Umschreibung Ihrer spontanen Vorstellung aufschreiben.



____

Bild: schlecht erkennbar -3 -2 -1 0 +1 +2 +3 gut erkennbar

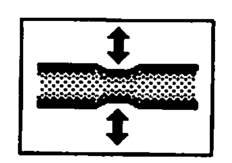
Ästhetik: unsympathisch -3 -2 -1 0 +1 +2 +3 sympathisch



____

Bild: schlecht erkennbar -3 -2 -1 0 +1 +2 +3 gut erkennbar

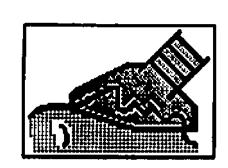
Ästhetik: unsympathisch -3 -2 -1 0 +1 +2 +3 sympathisch



____

Bild: schlecht erkennbar -3 -2 -1 0 +1 +2 +3 gut erkennbar

Ästhetik: unsympathisch -3 -2 -1 0 +1 +2 +3 sympathisch



____

Bild: schlecht erkennbar -3 -2 -1 0 +1 +2 +3 gut erkennbar

Ästhetik: unsympathisch -3 -2 -1 0 +1 +2 +3 sympathisch

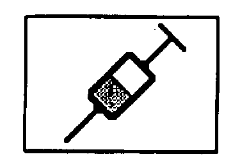


Bild: schlecht erkennbar gut erkennbar
-3 -2 -1 0 +1 +2 +3

Ästhetik: unsympathisch sympathisch
-3 -2 -1 0 +1 +2 +3

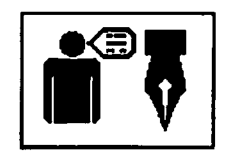


Bild: schlecht erkennbar gut erkennbar
-3 -2 -1 0 +1 +2 +3

Ästhetik: unsympathisch sympathisch
-3 -2 -1 0 +1 +2 +3

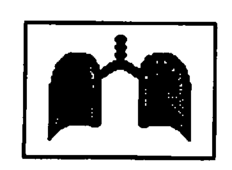


Bild: schlecht erkennbar gut erkennbar
-3 -2 -1 0 +1 +2 +3

Ästhetik: unsympathisch sympathisch
-3 -2 -1 0 +1 +2 +3

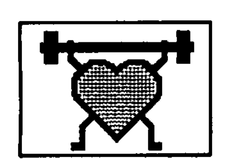


Bild: schlecht erkennbar gut erkennbar
-3 -2 -1 0 +1 +2 +3

Ästhetik: unsympathisch sympathisch
-3 -2 -1 0 +1 +2 +3

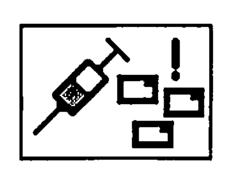


Bild: schlecht erkennbar gut erkennbar
-3 -2 -1 0 +1 +2 +3

Ästhetik: unsympathisch sympathisch
-3 -2 -1 0 +1 +2 +3

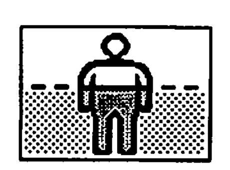


Bild: schlecht erkennbar gut erkennbar
-3 -2 -1 0 +1 +2 +3

Ästhetik: unsympathisch sympathisch
-3 -2 -1 0 +1 +2 +3

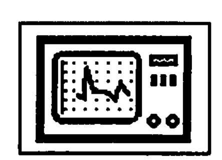


Bild: schlecht erkennbar gut erkennbar
-3 -2 -1 0 +1 +2 +3

Ästhetik: unsympathisch sympathisch
-3 -2 -1 0 +1 +2 +3

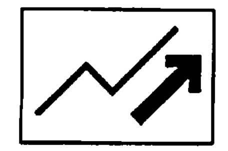


Bild: schlecht erkennbar gut erkennbar
-3 -2 -1 0 +1 +2 +3

Ästhetik: unsympathisch sympathisch
-3 -2 -1 0 +1 +2 +3

Bild: schlecht erkennbar gut erkennbar
-3 -2 -1 0 +1 +2 +3

Ästhetik: unsympathisch sympathisch
-3 -2 -1 0 +1 +2 +3

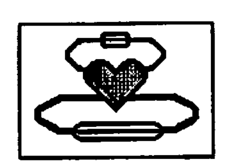


Bild: schlecht erkennbar gut erkennbar
-3 -2 -1 0 +1 +2 +3

Ästhetik: unsympathisch sympathisch
-3 -2 -1 0 +1 +2 +3

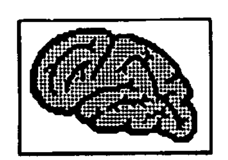


Bild: schlecht erkennbar gut erkennbar
-3 -2 -1 0 +1 +2 +3

Ästhetik: unsympathisch sympathisch
-3 -2 -1 0 +1 +2 +3

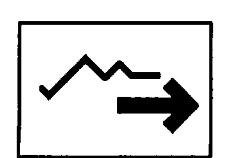


Bild: schlecht erkennbar gut erkennbar
-3 -2 -1 0 +1 +2 +3

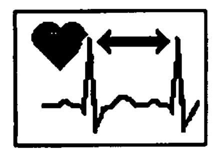
Ästhetik: unsympathisch sympathisch
-3 -2 -1 0 +1 +2 +3

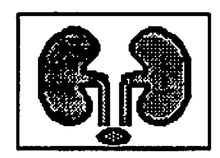
TEIL II

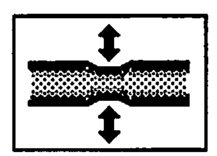
Im ersten Teil dieses Fragebogens konnten Sie Begriffe Ihrer eigenen Wahl zuordnen. Jetzt wird zu denselben Piktogrammen eine Liste von Begriffen vorgegeben, die den Piktogrammen zugeordnet werden. Hierfür muß für jedes Piktogramm die Liste einmal durchgesehen werden. Die Kennzahl des Begriff, der Ihrer Meinung nach am besten paßt, soll unter das entsprechende Piktogramm geschrieben werden. Ein Begriff kann mehr als einem Piktogrammen zugeordnet werden.

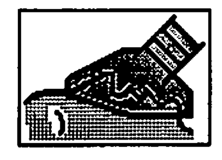
Begriffsliste:

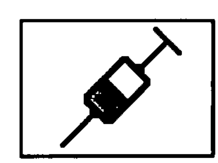
(1) Patientendaten (Anamnese)
(2) Patient-Monitoringgrößen
(3) Therapie
(4) Therapiebegründung
(5) Liste durchgeführter Therapien
(6) Herzfrequenz
(7) Intravasalvolumen
(8) Gefäßtonus
(9) Herzleistung/Kontraktilität
(10) Narkosetiefe
(11) steigender Trend
(12) Trend unverändert
(13) kardiovaskuläres System
(14) respiratorisches System
(15) Zentralnervensystem
(16) Nieren-System

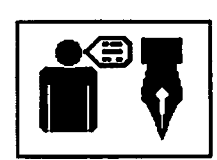

____

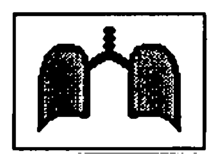

____


____

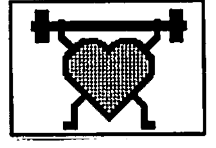

____

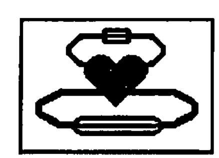

____

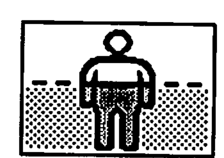

____

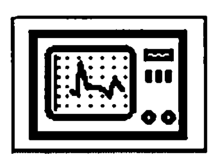

____

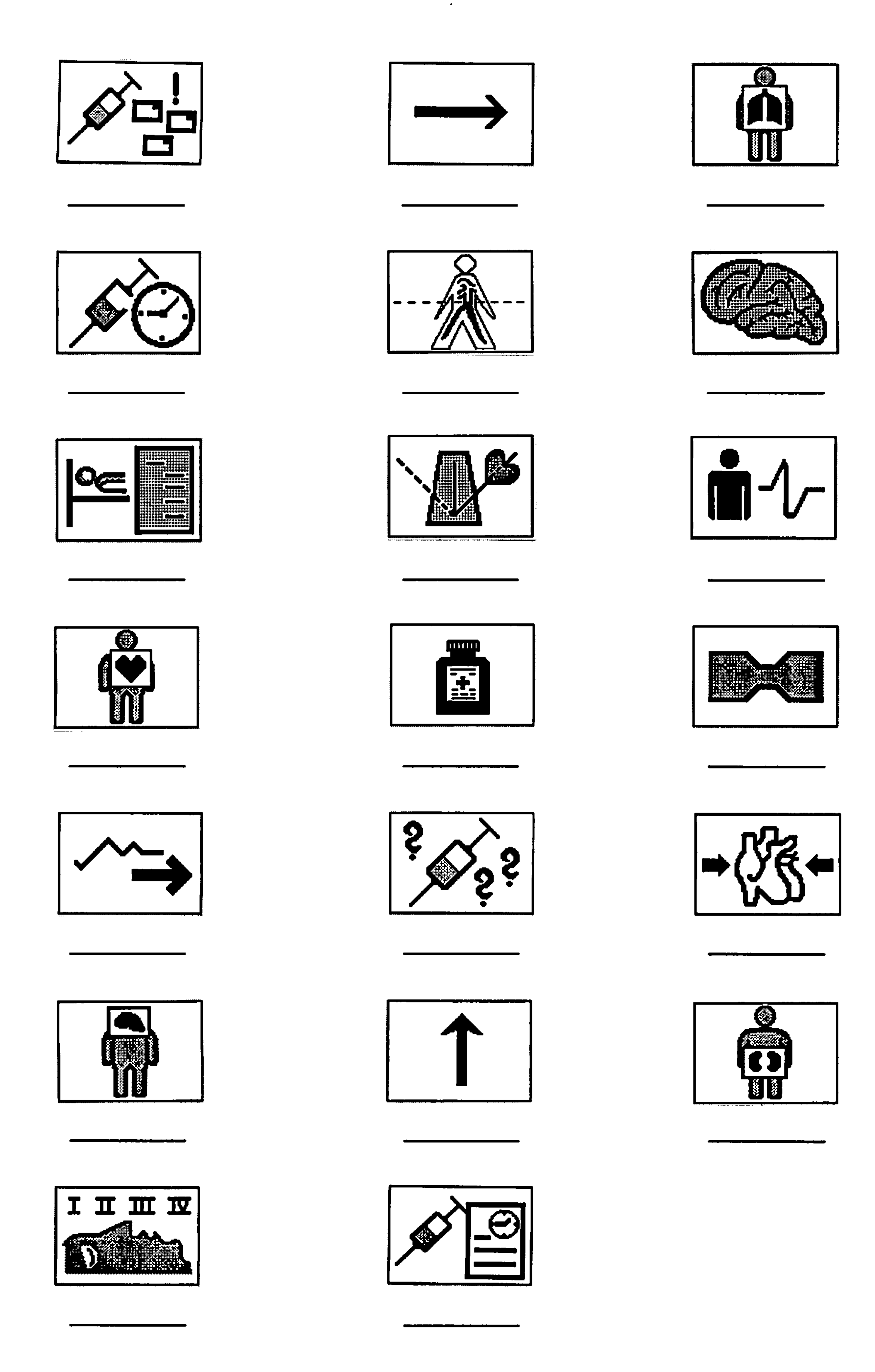

____


____


____

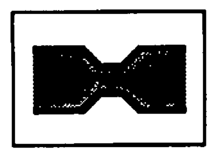

____


____


I II III IV

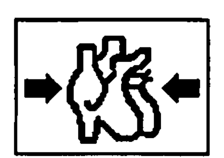
TEIL III

Der letzte Teil zeigt Ihnen eine mögliche Zuordnung von Begriffen und Piktogrammen. Prüfen Sie, inwieweit Ihnen die Piktogramme charakteristisch für die jeweiligen Begriffe erscheinen. Kommentieren Sie (falls erwünscht) diese Zuordnung und Darstellung.



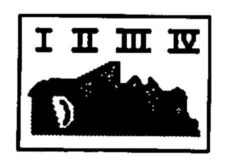
nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

Gefäßtonus



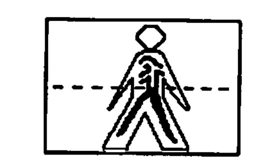
nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

Herzleistung/Kontraktilität



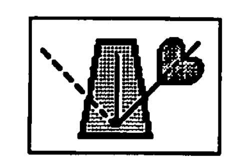
nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

Narkosetiefe



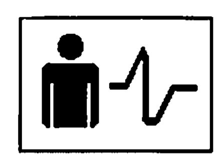
nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

Intravasalvolumen



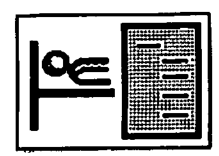
nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

Herzfrequenz



nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

Patient-Monitoringgrößen



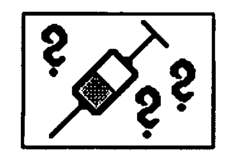
nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

Patientendaten (Anamnese)



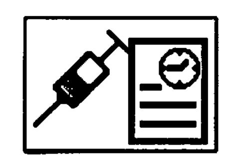
nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

Therapie



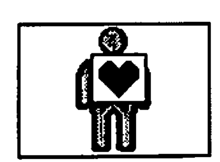
nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

Therapiebegründung



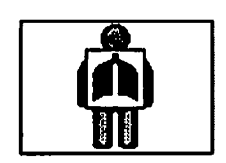
nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

Liste durchgeführter Therapien



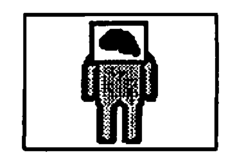
nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

kardiovaskuläres System



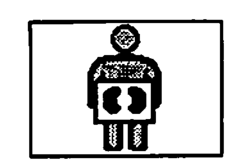
nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

respiratorisches System



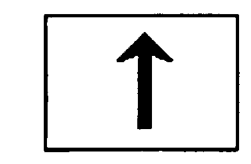
Zentralnervensystem

nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3



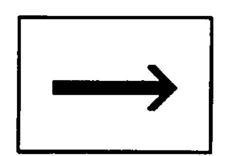
Nieren System

nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3



steigender Trend

nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3



Trend unverändert

nicht charakteristisch charakteristisch
-3 -2 -1 0 +1 +2 +3

---

2. WIE EMPFANDEN SIE DIESEN TEST?

unangenehm angenehm
-3 -2 -1 0 +1 +2 +3

uninteressant interessant
-3 -2 -1 0 +1 +2 +3

Vielen Dank für Ihre Mitarbeit!

A.2 Auswertung der Fragebögen

Die Auswertung der Piktogrammstudie wird separat für die Teilnehmerdaten und die Bereiche der "physiologische Teilsysteme", "Zustandsgrößen", "Informationsfenster" und "Trenddarstellung" vorgestellt.

A.2.1 Teilnehmer

Berufsgruppe	Ärzte	Pflegepersonal		PJ-Studenten
	15	11		4
Altersgruppe	bis 30 J.	31 - 40	41 - 50	männlich: 21
	16	11	3	weiblich: 9

Der Mittelwert der Einschätzung "Ich halte Piktogramme für
nicht sinnvoll - sinnvoll": 1.87.

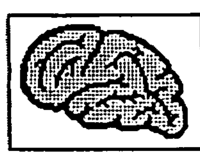
A.2.2 Piktogramme physiologischer Teilsysteme

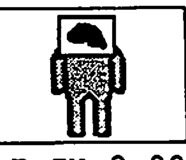
Die Markierung der t-Werte durch Sterne (*) kennzeichnet das Signifikanzniveau für die Hypothese, daß die Mittelwerte von zwei Bewertungen verschieden sind:
* => $p < 0.05$ ** => $p < 0.01$ *** => $p < 0.001$

Charakteristik	2.1	0.6	t= 2.56*
Erkennbarkeit	0.8	2.1	t=-2.61*
Ästhetik	0.4	1.5	t=-2.16*
spontan identif.	0.70	0.30	
richtig zugeordnet	0.87 n.zu.0.07	0.57 n.zu.0.10	
verwechselt mit	9(2x)	6(7x), 9(3x)	

Charakteristik	2.9	2.2	t= 3.03**
Erkennbarkeit	1.6	2.1	t=-1.01
Ästhetik	1.1	1.4	t=-0.54
spontan identif.	0.90	0.40	
richtig zugeordnet	1.00 n.zu.___	0.97 n.zu.0.00	
verwechselt mit		2	

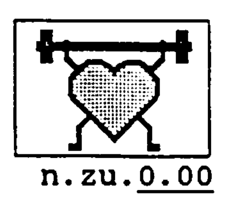
Charakteristik	2.9	2.2	t= 3.03**
Erkennbarkeit	1.7	0.1	t= 2.60*
Ästhetik	1.4	1.2	t= 0.42
spontan identif.	0.90	0.50	
richtig zugeordnet	1.00 n.zu.___	0.97 n.zu.0.00	
verwechselt mit		7	





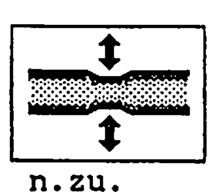
Charakteristik	2.8	1.5	t= 3.52**
Erkennbarkeit	1.2	0.4	t= 1.14
Ästhetik	0.4	-0.4	t= 1.18
spontan identif.	0.80	0.30	
richtig zugeordnet	0.97 n.zu.0.03	1.00 n.zu.0.00	
verwechselt mit			

A.2.3 Piktogramme der Zustandsgrößen

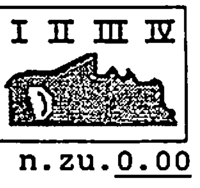


Charakteristik	2.3	0.7	t= 3.88***
Erkennbarkeit	1.4	-0.8	t= 4.26***
Ästhetik	1.7	-0.5	t= 3.79***
spontan identif.	0.90	0.20	
richtig zugeordnet	0.97 n.zu.0.00	0.63 n.zu.0.07	
verwechselt mit	13	13(8x), 8	

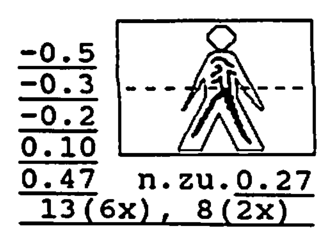
Charakteristik	1.9	1.1	t= 1.11
Erkennbarkeit	1.2	0.3	t= 1.25
Ästhetik	0.7	1.0	t=-0.44
spontan identif.	0.90	0.60	
richtig zugeordnet	0.90 n.zu.0.00	0.77 n.zu.0.10	
verwechselt mit	9(2x), 2	9(3x), 13	



Charakteristik	2.3	0.7	t= 2.49*
Erkennbarkeit	0.8	0.5	t= 0.43
Ästhetik	0.6	0.5	t= 0.26
spontan identif.	0.50	0.50	
richtig zugeordnet	1.00 n.zu.___	0.90 n.zu.0.03	
verwechselt mit		7, 9	



Charakteristik	2.3	-0.9	t= 6.96***
Erkennbarkeit	0.4	-0.1	t= 0.62
Ästhetik	0.8	-0.3	t= 1.60
spontan identif.	0.70	0.20	
richtig zugeordnet	0.97 n.zu.0.00	0.80 n.zu.0.17	
verwechselt mit	2	14	



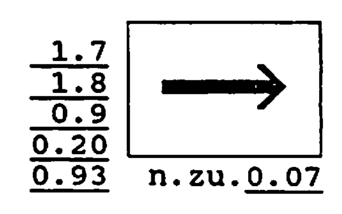
Charakteristik	-1.8	-0.5	t=-2.26*
Erkennbarkeit	-1.5	-0.3	t=-2.01
Ästhetik	-0.2	-0.2	t= 0.0
spontan identif.	0.10	0.10	
richtig zugeordnet	0.47 n.zu.0.33	0.47 n.zu.0.27	
verwechselt mit	3,3,10,10,8,11	13(6x), 8(2x)	

A.2.4 Piktogramme der Informationsfenster

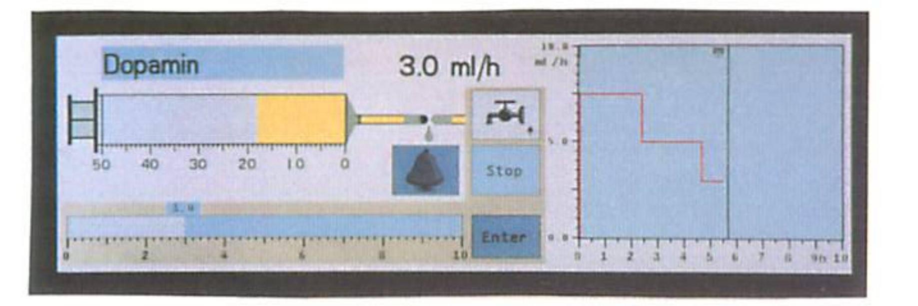
	Piktogramm A	Piktogramm B	t
Charakteristik	1.1	-1.0	t= 3.56**
Erkennbarkeit	1.9	0.4	t= 2.45*
Ästhetik	1.4	1.1	t= 0.69
spontan identif.	0.90	0.00	
richtig zugeordnet	0.93 n.zu.0.03	0.27 n.zu.0.33	
verwechselt mit	13	6(5x),9(4x),13(2x),10	
Charakteristik	0.9	0.9	t=-0.1
Erkennbarkeit	-1.1	-0.9	t=-0.36
Ästhetik	-0.3	-0.1	t=-0.31
spontan identif.	0.20	0.30	
richtig zugeordnet	0.53 n.zu.0.03	0.73 n.zu.0.13	
verwechselt mit	5(9x),2(2x),3,4	4(3x), 10	
Charakteristik	0.9	1.1	t=-0.3
Erkennbarkeit	1.8	0.9	t= 1.3
Ästhetik	1.0	1.0	t= 0.0
spontan identif.	0.70	0.60	
richtig zugeordnet	0.87 n.zu.0.07	0.90 n.zu.0.10	
verwechselt mit	5(2x)		
Charakteristik	-0.7	0.1	t=-1.15
Erkennbarkeit	-1.7	0.1	t=-2.53*
Ästhetik	-0.9	-0.1	t=-1.26
spontan identif.	0.10	0.00	
richtig zugeordnet	0.37 n.zu.0.03	0.77 n.zu.0.07	
verwechselt mit	3(3x),5(14x),13	3(5x)	
Charakteristik	0.8	-0.6	t= 2.12*
Erkennbarkeit	1.1	0.5	t= 0.82
Ästhetik	1.0	0.6	t= 0.68
spontan identif.	0.80	0.60	
richtig zugeordnet	0.63 n.zu.0.00	0.30 n.zu.0.13	
verwechselt mit	3(7x), 4(4x)	3(16x), 10	

A.2.5 Piktogramme für die Trenddarstellung

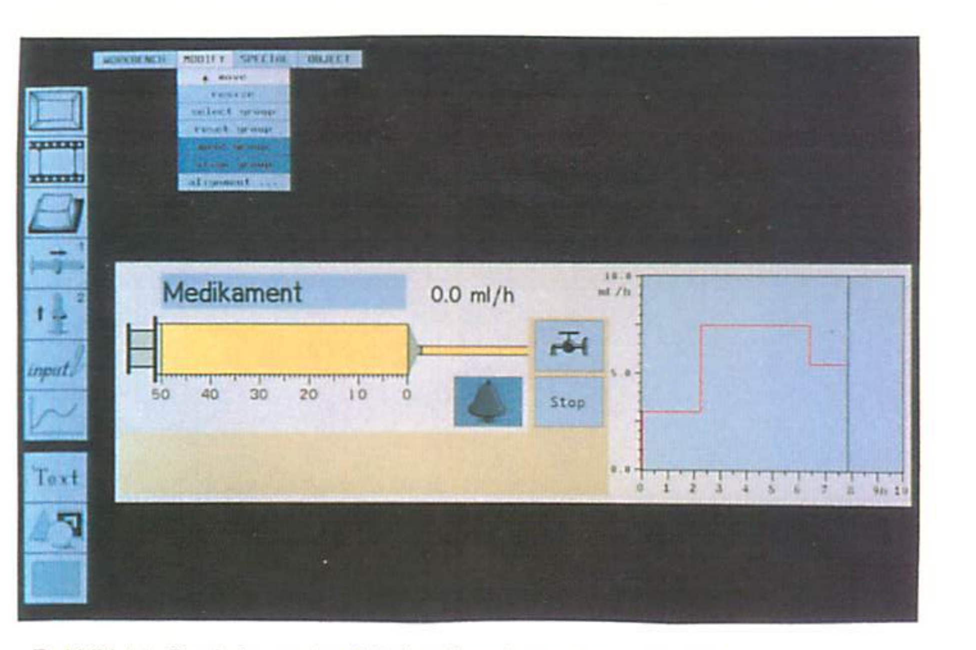
	Piktogramm A	Piktogramm B	t
Charakteristik	2.5	1.7	t= 1.44
Erkennbarkeit	0.1	2.2	t=-3.68***
Ästhetik	0.5	1.3	t=-1.44
spontan identif.	0.30	0.40	
richtig zugeordnet	1.00 n.zu.____	0.97 n.zu.0.03	
verwechselt mit			
Charakteristik	2.1	1.7	t= 0.87
Erkennbarkeit	-1.1	1.8	t=-4.58***
Ästhetik	0.1	0.9	t=-1.71
spontan identif.	0.30	0.20	
richtig zugeordnet	0.97 n.zu.0.03	0.93 n.zu.0.07	
verwechselt mit			



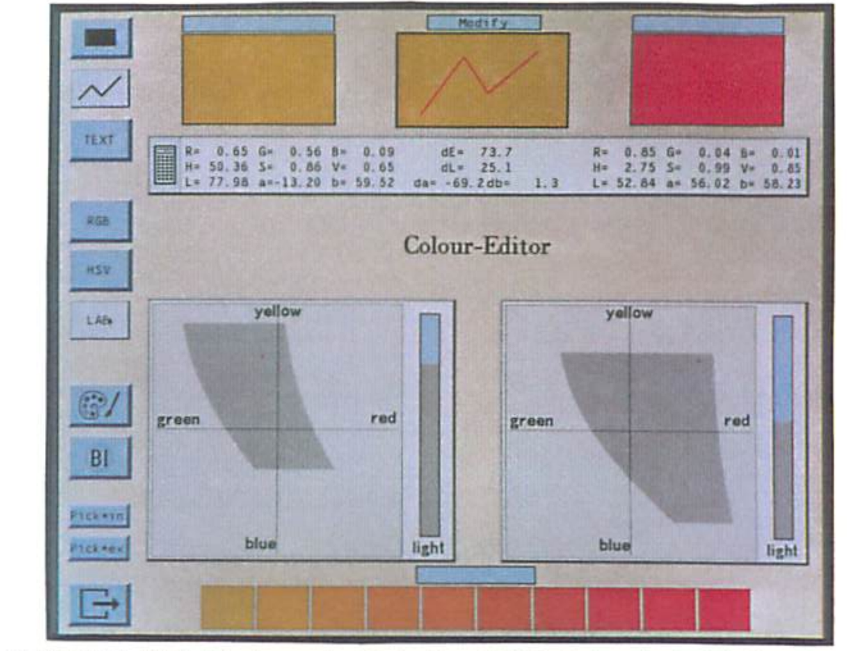
Anhang B: Farbbilder



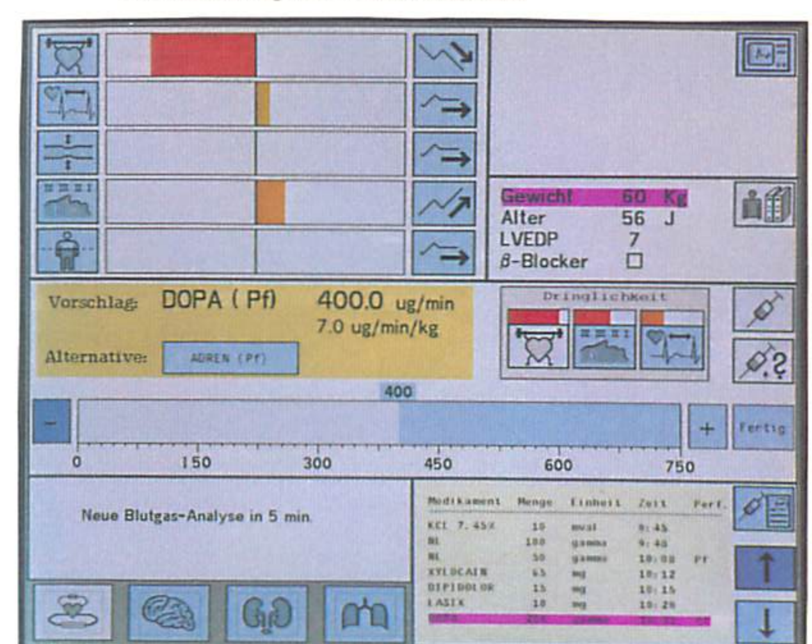
Farbbild 3.1: Infusionspumpen-Kontrolleinheit nach Aktivierung der Dosis-Taste.



Farbbild 4.1: Bearbeitung einer Schnittstelle mit dem Entwurfseditor.



Farbbild 4.2: Bildschirmlayout des Farbeditors CEDI mit Angabe der Farbkoordinaten und Darstellung im CIELAB-Farbraum.



Farbbild 6.1: AES-2 Bildschirm: Erklärungskomponente der Therapieempfehlung.